Faculty of Arts, Environment and Technology

Leeds Metropolitan University

Investigation into Domestic Energy Consumption

GreenAPP- Energy Monitoring Mobile Application

Author: Sophie Greene

Submitted in partial fulfilment of the requirements for the

MSc Web Application Development

September, 2011

Abstract

Carbon emission reduction is one of the most talked about issues in recent times. A great deal of research has been conducted in this area and a lot more needs to be done. Therefore green computing has become one of the hot topics in recent years by several companies as well as the governments.

This research investigates the use of persuasive technology such as ubiquitous computing and its role in tackling energy consumption behaviour by providing an intuitive mobile application (GreenAPP) to monitor the usage of energy in domestic environment.

GreenAPP is an Android based mobile application which was developed based on consultations with potential users. Furthermore a data transmission embedded device was setup in order to achieve data ubiquity. These gadgets were used alongside other research methods explore the effect of making energy consumption visible on the overall energy consumption behaviour of the users. Results are encouraging; showing an overall drop in electricity use of participating households.

Acknowledgements

To Jon Beech: I am grateful for you friendship and help

To Muthu Ramachandran: I appreciate your supervision, advice and help

To Glenna Larsen: Thank you for being an optimist, I would not have finished this without you

And to

David Moore, Duncan Mullier, Patrick Ingham, Meg Soosay

Mark Dixon, Julian Old and Simon Sharpe

For the advice and the encouragement

To my research participants

Thank you. This would not have been possible without you

To William: your mom is back

Table of Contents

Figures and Tables

Definition of Terms

Human Computer Interaction (HCI)

Human-Computer Interaction (HCI) is defined as "the study and practice of usability" (Carroll, 2001, p. xxvii). Carroll ibid claims that the tools that promote, perform and measure usability are benchmarks in the computing culture.

Interaction Design

Interaction Design can be defined as the comprehensive relationship between designed applications, their users and the context within which the interaction takes place. The main focus in the field of interaction design is "the user experience". Fallman (2008)

Global Warming

Global warming is the increase of the average earth temperature as a result of human activity related greenhouse gases emission (Conway, 2008)

Sustainable Development

Sustainability or sustainable development is "development which meets the needs of the present without compromising the ability of future generations to meet their own needs." (The World Commission on Environment and Development, UN, 1987)

Internet of Things

The Internet of Things (IoT) is a networked interconnection of objects (including users), which enables objects to interact with each other. It extends the Internet into the physical world enabling objects to be managed remotely and act as physical access points to Internet services. IoT is established on the ground of the proliferation of wireless sensor network, MobiComp (mobile computing), UbiComp (Ubiquitous Computing) and information technologies.

Embedded Systems

An embedded system is a computing system designed to perform a specific task. Examples are mp3 players.

Abbreviations

ABBREVIATION	STANDS FOR
API	Application Programming Interface
CSV	Comma Separated Values
DEHEMS	Digital Environmental Home Energy Management System
GREENAPP	Energy Monitoring Application
HCI	Human Computer Interface
I/O	Input/ Output
IDE	Integrated Development Environment
IOS	iPhone OS
IOT	Internet of Things
KGCO2E	Kilogram of Carbon Dioxide equivalent
KWH	Kilo-Watt hours
PCB	Printed Circuit Board
REST	Representational State Transfer
UML	Unified Modelling Language
WSN	Wireless Sensor Network
XML	Extensible Mark-up Language

Chapter 1 Introduction

This project's outcome is GreenAPP; an Android based mobile application tailored to reduce energy invisibility and attempt to change energy consumption behaviour. Research methods are used throughout the development process in order to inform design decisions and demonstrate that the product meets its purpose.

This chapter gives a brief overview of the motivation behind conducting this project and outlines the key aims and objectives. Finally, an overview of the chapters of this report is presented.

1.1 Motivation

Global warming, climate change, carbon emission, greenhouse gasses, are terms we are all familiar with. The mass media coverage, government targets and international attention has brought these terms to the public eye. Despite the debate surrounding the causes of global warming, it is widely accepted that the human race is contributing to it (Mastrandrea & Schneider, 2005). It is also widely accepted that humankind has a responsibility not only towards the current generation but towards future generations to do what they can to try and reduce the overall carbon footprint as well as sustain natural resources such as oil and gas for generations to follow. Statistics from the Department of Energy & Climate Change (2009) demonstrate that the expenditure on energy for domestic use in 2009 accounts for almost 20% of the total expenditure of the UK on energy. This shows that energy saving in the household is a key factor when looking at reducing the nation's carbon emission.

This is a product-based research project. The idea came about through a literature review into pervasive technology applications and an interest in the environment and green issues. The project aims to answer the following research question:

"Can a persuasive mobile application be designed to change users' behaviour regarding electricity consumption?"

1.2 Aims and Objectives

The project aims to investigate the use of ubiquitous technology such as mobile devices in a persuasive domain. A mobile application which monitors and displays summative information about the different appliances and the overall household electricity usage has been developed. Finally, the effect of using such an application on the household's electricity usage is explored through investigating the public views on the matter. This is achieved through the following:

- Conduct an extensive literature review in the fields of persuasive technology, green energy, household energy consumption, pervasive/ ubiquitous computing and mobile application development including interaction design and HCI guidelines.
- Design, implement, conduct and analyse an online survey aimed at computer/internet users. The survey investigates participants' sustainable development awareness, incentives for energy consumption reduction, willingness to use energy monitoring devices, preferred feedback format and preferred design interface.
- Design, implement, test and evaluate a mobile application which receives and displays real time electricity readings from the user's household energy monitor. The mobile application tackles energy invisibility as well as encouraging users to save energy. The outcomes of the online survey mentioned in the previous point, such as user preferences alongside interaction design guidelines and HCI principles is used to aid the design and development of the application
- Design, conduct, implement and evaluate an interview / trial combination. This combination is used to test the effects of using the mobile application on a sample of Yorkshire area residents.
- The research project as a whole alongside the effectiveness of the chosen research methodology and the developed application is evaluated using recognised evaluation techniques.

1.3 Dissertation Structure

The remainder of this dissertation starts with a survey of relevant literature in chapter 2. Chapter 3 outlines the methodology used to conduct this research including the

research methods used, the tools and devices used to conduct the trials, the data transmitter hardware and the mobile application design and the rational for choosing to develop for the Android platform. Chapter 4 describes the software implementation process for the data transmitter (GreenAPP Transmitter) as well as the mobile application implementation. Chapter 5 presents the findings of the research including an analysis of the online survey and the trials. The first part of chapter 6 includes the illustration and description of the techniques used to validate and test the mobile application and the data transmitter. The second part of the aforementioned chapter presents a reflective evaluation of each phase of the research project as well as an evaluation of the overall research process. Finally, chapter 7 gives the dissertation conclusions and illustrates possible future directions.

Chapter 2 Literature Review

This chapter reviews the areas of research that inspired and supports the work in this project. The first part of this chapter introduces these areas of research. The second and concluding part of this research introduces the main issues addressed by researchers in these fields such as energy invisibility, security issues and data transmission.

2.1 Areas of Research

2.1.1 Domestic Energy Consumption Research

In the last 5 years, the research in the field of domestic energy consumption has increased dramatically. Domestic energy consumption research is a field of research concerned with the motivating factors behind energy saving in the household. According to Hazas et al. (2011) the domestic energy consumption (mainly electricity consumption) research began in the 1970's in the USA and parts of Europe fuelled by a desire for energy independence resulting from the oil crisis at that time. The increased focus on acid rain awareness in the 1980s kept this field of research alive even after oil prices fell down. The emergence of global warming and the public attention it received since the 1990s to date has inspired researchers to continue working in the field. Since the beginning of the twenty-first century, global warming became part of a more general publication i.e. sustainable development. So for over forty years, the researchers in the field of domestic energy consumption were motivated by different factors. Regardless of motivation, the branches of research in this field can be categorised into three main classifications which are basic opinion survey results, energy consumption modelling and intervention techniques such as feedback and incentives. The latter is the focus of this research.

2.1.2 Captology- Persuasive Computing

Persuasive computing or captology according to Fogg (1997) is the study of computing devices as persuasive tools. Although the term captology was introduce by Fogg in 1996, persuasive computing have always periphery in many other fields such as Human Computer Interaction HCI and Interaction Design.

2.1.3 Ubiquitous Computing

Ubiquitous Computing or Ubicomp also known as Pervasive Computing is a term introduced by Mark Weiser in 1988. Weiser described computers as being invisible. Ubiquitous computing enables users to achieve their aims without being confided to one place. The advancement in mobile technology and the introduction of the third generation of mobile phones has led to a pervasive revolution (Saffer 2009). Designing user interface for pervasive devices such as mobile phones is more challenging to designers than the design process for desktop application. The designer needs to be aware of the context of use, the physical location of the user and the environment surrounding them.

2.2 Energy Invisibility

Tackling energy invisibility is the main concern of research in the area of household energy consumption. Users are not aware of the amounts of energy they are consuming at a specific instance, they are only aware of using appliances and devices therefore energy monitoring or feedback could be the key to encouraging individuals to save energy.

Some research has been conducted into feedback display designs and individual appliance monitoring. Fitzpatrick & Smith (2009) conducted a study into energy monitoring display units which showed that people in the home environment are interested in getting feedback on their energy use. Providing detailed consumption figures for each appliance in the household helps users investigate the peaks in their usage which could affect future consumption. Ibid also express a set of design concerns that identify and concentrate on current research into the user experience and guidelines to produce effective and usable feedback display designs. Chao et al (2010) emphasised the importance of reducing energy invisibility and managing household energy consumption using appliance profiles as well as consumption history. Although the study focused on displays that are provided by the energy provider, the study is important in laying down guidelines to developing interactive and pervasive mobile applications, which can act as monitors of energy consumption.

Chetty et al. (2008) proposed the use of UbiComp to achieve sustainability through tackling the invisibility of energy consumption. Ibid conducted a qualitative study into householders attitude towards energy saving. The study showed that people are already modifying their homes through insulation, energy efficient appliances etc. in order to reduce energy consumption. The study focused on finding out the quality and quantity of information users are interested in seeing when looking at energy consumption displays. Users were keen on, at a glance, monetary information as it was very indicative of the financial impact of their consumption patterns. Participants were also interested in more detailed information which enabled them to analyse their consumption and rectify it. Chetty also highlighted the challenge of green data collection

2.3 Security and Connectivity

Making energy readings available for a mobile application involves transmitting data collected by sensors in the home environment over the internet either directly to the mobile application or to an intermediate server. This introduces the security risks involved when using ubiquitous context aware applications. The data collected by an energy monitor in a household and sent over the internet can fall into the hands of potential intruders which can lead to potential exposure of sensitive details about the household; for example, energy consumption decreases when people are on holiday.

Using ubiquitous devices as a pervasive persuasive tool requires employing a data communication protocol or method. This is an extensive field of research and includes different branches such as network communication, Wireless Sensors Network (WSN) and Service Oriented Device Architecture SODA. SODA is the standard used to govern interaction between embedded devices and web services. It treats devices as services and enables integration of devices in business.

Shah et al. (2009) introduce some of the challenges involved in collecting and transmitting sensor monitoring data through their work on the DEHEMS project. DEHEMS a home energy management system; uses embedded devices and web services standards for interaction of devices and users with the server.

Bos et al (2007) address the security issue involving data communication when developing pervasive domestic energy consumption applications by applying and extending the Javacard As Secure Objects Networks JASON architecture.

2.4 Related work

Spagnolli et al. (2011) developed an Apple iOS game application "Energy Life" to monitor energy consumption in the household environment and monitored the effect of using it on participant's consumption. ibid found that using Energy Life caused a 5% decrease in the household's consumption. The application in ibid relies totally on environmental impact and actual use to feedback, leaving the use of monetary value feedback unexplored.

2.5 Summary and Conclusion

The main themes of research in the field of household energy consumption have been identified. Some work in the field has been carried out for energy monitoring displays. The core issue has been identified, which is the invisibility of energy consumption. Other issues are security and connectivity. Very little work has been done on live single appliance feedback apart from Spagnolli's EnergyLife for iOS. This is where GreenAPP fits; it attempts to fill the gap in the research regarding single appliance consumption.

Chapter 3 Methodology and Design

This chapter introduces the elements involved in the aforementioned process. The first part of this chapter sets out the research approach to this project. The second part lays out the system architecture including any devices or hardware used and the final part presents the UI design of the mobile application as well as the data transmitter.

3.1 Research Methodology

The research project involves developing an artefact of a persuasive nature. Researchers in the field of persuasive technology tend to conduct initial investigations into the likelihood that users will want to use technology and user preferences regarding the design of the system. The system is then developed taking into account the data collected in the previous stage. Finally, trials are conducted to investigate the effect of using the system on the users' behaviour under consideration (Fogg, 2009). Froehlich et al. (2009) used online survey and field study to aid the development of a persuasive transport application. Figure 3-1 presents the model of progression for this project.

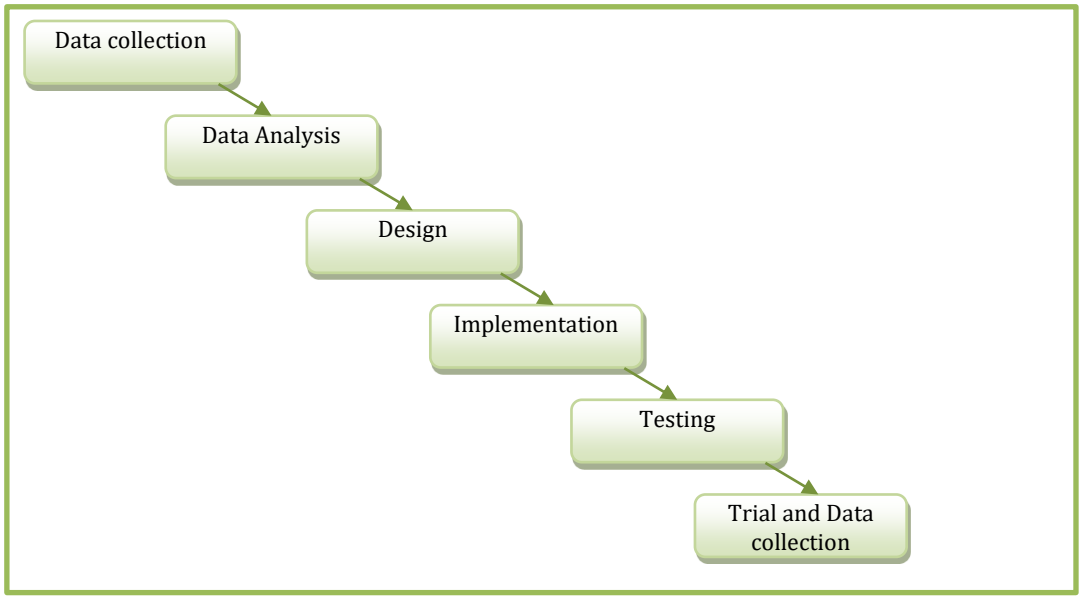


Figure 3-1 Project Flow Model

3.1.1 Mixed Research methods

This project uses the principle of triangulation in order to enhance the outcomes. A variety of research approaches and methods will be used to answer the research question. There are many reasons for combining different research methods to deliver

one research project including triangulation, complementary, development, initiation and expansion (Bryman, 2006).

This project deploys different research methods at specific stages for the following reasons

- Triangulation: the use of different research methods to converge, verify, correspond and triangulate the findings.
- Illustration: the use of qualitative data to illustrate quantitative findings
- Confirm and discover: an online survey is used to obtain information that is used in the mobile application design.
- Enhancement or building upon quantitative findings: The analysis of the online survey outcomes is used to funnel for more information and create a more useful interview.

3.1.2 Survey

Overview

O'Leary (2010) and Robson (2002) define surveying as the process of collecting quantitative data by asking individuals the same questions related to certain topics. Questionnaires are very powerful tools in collecting small amounts of data in standardised format from a large number of individuals.

GreenAPP's Survey

An online survey was designed to conduct a formative study aiming to investigate the respondents' willingness to shift to more eco-friendly domestic energy consumption behaviours and patterns, their motivations for saving energy, and reactions to different visual representations of energy consumption. The information gathered through the literature review stage of this project acted as guidelines when formulating the survey questions. Three main types of information were identified

- Demographical information: collected to aid the analysis of the rest of the information through categorisation; for example place of residence, household number of residents and so on

- ❖ Ideological information: collected to help form an understanding of what is important to different people based on their view of the world; for example environmental awareness and how that affects the type of feedback the participant responds to.
- ❖ Opinions and preferences: preferred themes of feedback, type of information participants respond to.

More details on the implementation, findings and analysis of the online survey can be found in Chapter 5.

3.1.3 Interviews

Overview

According to O'Leary (2010) an interview can be defined as a method of qualitative data collection which entails researchers pursuing open-ended answers related to a topic area. Fogg (2009) advises researchers in the field of persuasive technology to investigate the reasons that prevent the target behaviour and sum it into three categories: lack of motivation, lack of ability or lack of a well-timed prompt to carry out the behaviour. The reason could also be a combination of the three.

GreenAPP Interviews

Following Fogg's advice, the interviews were designed to collect qualitative data about energy consumption behaviour of the participants. Although an open questioned questionnaire could be used to collect the same data, it is important to conduct interviews because they allow rapport and trust to be developed between the researcher and the participant. It also allows the researcher to pick up on non-verbal cues and explore digressions. More about the implementation, analysis and findings of the interviews can be found in Chapter 5.

3.1.4 Observation

Overview

Observation is a method of indirect data collection which relies on the researcher's ability to gather information through their senses (O'Leary, 2010).

GreenAPP's Observation

A non-participant, candid, structured observation will be used to monitor participants' energy consumption through the use of through having access to participants secure remote server details.

3.1.5 Experiments

Overview

An 'experiment' is defined by Robson as "a research strategy characterised by the researcher actively manipulating or changing aspects of what is studied" (Robson, 2002, p 547). An experiment can also be defined as an empirical investigation under controlled conditions designed to examine the properties of, and relationship between, specific factors (Denscombe, 2007).

GreenAPP's Trials

No experiment in its ideal definition was designed for this project. However, a trial or a field study was designed to investigate the effect of installing an energy monitoring system in the participating households and using it alongside the GreenAPP mobile application.

3.1.6 The overall Methodology

A combination of interviews, observations and trials was designed to investigate the effects of using GreenAPP on a sample of Leeds City residents. The combination is in the following order

- Record pre- trial energy consumption
- Conduct pre-trial Interview
- Deploy the energy monitoring system and GreenAPP
- Observe consumption throughout the system deployment period
- Conduct post-trial interviews

3.2 Energy Monitoring System's Architecture

In order to provide GreenAPP with the electricity reading pervasively; the electricity readings needs to be taken recorded and sent over the internet. This section describes the

devices, tools and hardware used in the aforementioned process. Figure 3-2 *illustrates the overall system installed in the home environment.*

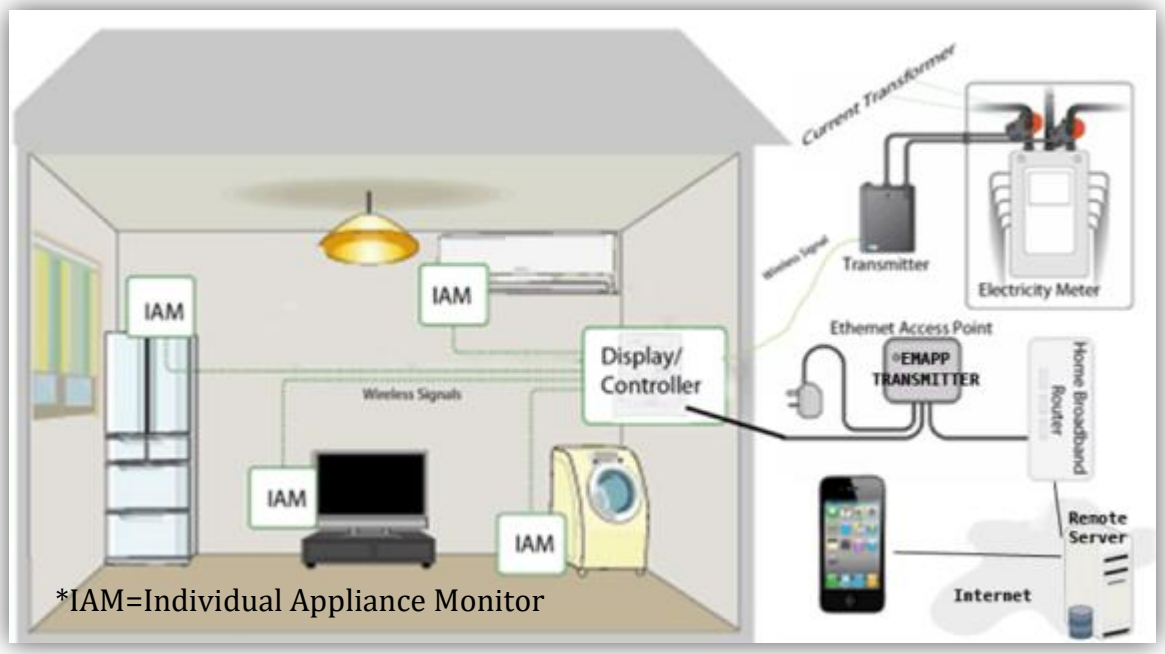


Figure 3-2 Consumption Monitoring System

3.2.1 Energy Monitoring Technology

Overview

There are a number of "off the shelf" energy monitoring devices provided by different vendors. In the past five years, monitors have gained popularity. A large number of products offer consumers low-cost energy monitoring and feedback such as monitors supplied by CurrentCost, OWL, Efergy and Unifi.

Principle of Operation

The electricity meter current reading, which represents the overall household consumption at a specific instant of time, is acquired using electromagnetic flow sensing coil also known as a current transformer CT. The reading is then transmitted wirelessly to a central device or a display. Appliance energy consumption is acquired through a smart plug or an Individual Appliance Monitor IAM which is a wireless electricity sensor that acquires the instantaneous electricity reading and transmits it to the main display unit.

The Wireless technology specification behind the communication between the main controller/display, the CT and the IAMs (Individual Appliance Monitor) and the main transmitter is known as ZigBee specification. The devices communicate as a Wireless Sensor Network WSN. The energy monitor used to conducts the trials of the GreenAPP is the CurrentCost CC128 Devices Used

Figure 3-3 Energy monitoring devices used in GreenAPP trials

In addition to reading the current appliance or electricity meter's power measured in Watt. It measures the current room temperature and the display unit also shows some historical usage data.

3.2.2 Data Transmission and Storage

Overview

GreenAPP is a pervasive application, in order to function; it needs the current electricity readings. GreenAPP could handle the need to access remote data in two ways; receive the data directly from the household which requires massive storage space and processing power which a smartphone on a limited battery life can never handle. The other and more sensible way is to read the data from a web server dedicated for receiving data from the household, storing it, and sending it to the smartphone upon request. Whichever of the aforementioned methods is used the data collected by the energy monitoring system needs to be transmitted to a remote destination over the internet.

The Ethernet Access Point, available off-the-shelf, transmits data for the main meter sensor only and not for the appliances. To overcome this problem and for the purpose of this project, an embedded device was built and programmed to read the data from the monitoring device and transmit it in XML format to the remote web server or platform.

Data Storage

The remote destination chosen to store data for the implementation of GreenAPP is a third party data brokerage platform stores, converts and serves data in multiple data formats including XML and CSV. The platform provides most of its functionality via its API and is entirely based on HTTP requests, and conforms to the design principles of Representational State Transfer REST.

Data Transmitter Hardware

Arduino USB Board

This consists of a microcontroller, I/O pins and USB connection. The latter allows connections to a computer for programming and debugging purposes via a special Arduino IDE. The board's capabilities can be extended using different "shields", in the case of the data transmitter an Ethernet Shield provides the needed internet connectivity to communicate with the server. Figure 3-4 and Figure 3-5 show the printed circuit board PCB of the main board and the Ethernet shield respectively

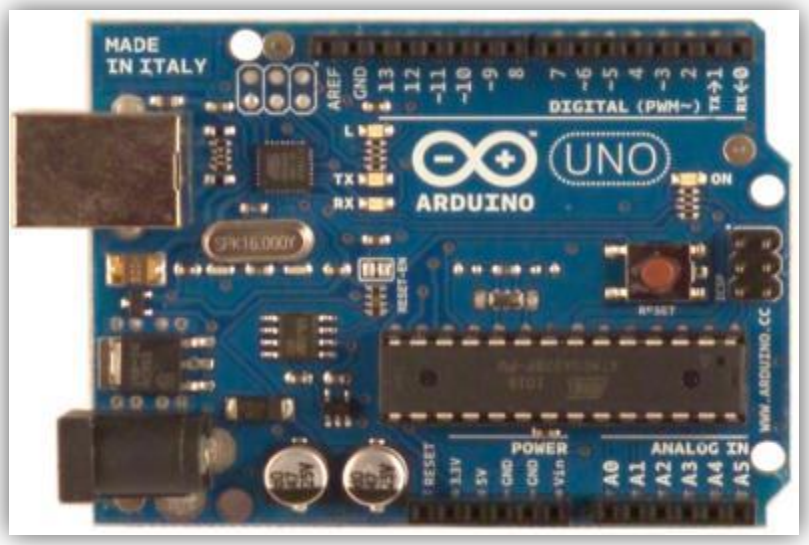


Figure 3-4 Arduino Board (Arduino, n.d.)

Figure 3-5 Arduino Ethernet Shield (Arduino, n.d.)

Figure 3-6 shows the actual deployment of the GreenAPP transmitter and its connection to the main display of the energy monitoring system.

Figure 3-6 Electricity Monitor connected to transmitter via serial cable

3.3 GreenAPP Mobile Application Design

GreenAPP is designed for the Android platform. The development process of a mobile application depends on the platform the application is built for. This section discusses the platform selection rationale and the methodology used in designing GreenAPP.

3.3.1 Why Android

Android is an attractive platform for mobile application developers because Google, the vendor, has made the most of the Android platform available under the Apache free-software and open source license which gives developers more freedom to build software for it. According to many market research companies Android is the fastest growing mobile operating system. In 2009, Android had 4% Smartphone OS North America's market share, compared to 15% for iPhone and 20% for BlackBerry. In 2010, Android's North America's market share exceeded that of the added share of the iPhone and Blackberry (Pinkasovitch, 2010). This information and a comparison of sales for the main Smartphone OS in North America over the 2009/2010 period are illustrated in Figure 3-7. In the third quarter of 2010 Android became the marker leader in North America and it is currently dominating 39% of the market share.

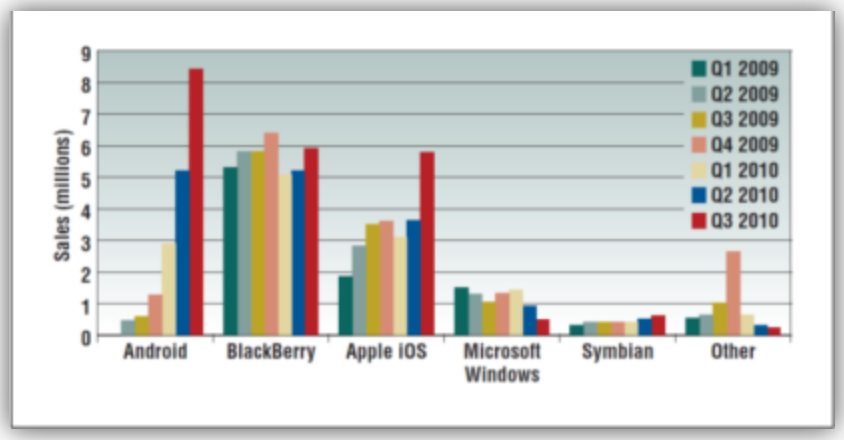


Figure 3-7 North America smart phone unit sales by quarter (Butler, 2011)

The same scenario applies to the worldwide market share as illustrated in Table 3-1 which compares the market shares of Smartphone OS vendors for the fourth quarter of 2009 and 2010 as well as their growth rate.

OS vendor	Q4 2010 shipments (millions)	% share	Q4 2009 shipments (millions)	% share	Growth Q4'10/Q4'09
Total	101.2	100.0%	53.7	100.0%	88.6%
Google	33.3	32.9%	4.7	8.7%	615.1%
Nokia	31.0	30.6%	23.9	44.4%	30.0%
Apple	16.2	16.0%	8.7	16.3%	85.9%
RIM	14.6	14.4%	10.7	20.0%	36.0%
Microsoft	3.1	3.1%	3.9	7.2%	-20.3%
Others	3.0	2.9%	1.8	3.4%	64.8%

Table 3-1 Smartphone's OS Vendors Market Share (Canalys, 2011)

The Android Software Platform

The Android operating system is built on a modified Linux kernel as shown in Figure 3-8. The software stack contains Java applications running on a virtual machine, and system components are written in Java, C, C++, and XML.

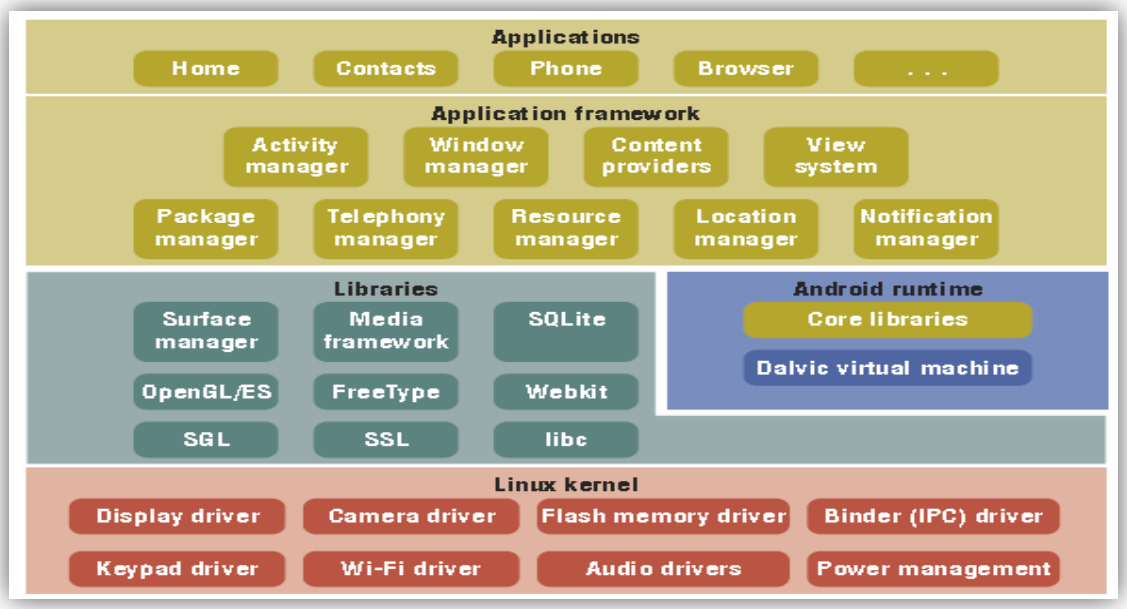


Figure 3-8 Android system architecture (Butler, 2011)

Applications and Developers

A smart phone's ability to complement our lives is directly related to the richness and quality of its applications. A key difference between Apple's App Store and Android Market is that the Android Market is open; the Apple App Store is guarded. That is, developers must submit applications for publication to Apple, and Apple decides what gets published, whereas developers self-publish to the Android Market. Google follows an unrestricted, open model for the Android Market. This is a significant advantage from a 'developer's point of view, which enables them to control publication; therefore, more applications will be available to consumers because publishing them is easier. This leads to consumers being inundated with low-quality applications, making finding high-quality applications more difficult. Google is aware of this problem and are investigating solutions to enable high-quality apps to rise to the top. GreenAPP is available on the Android market as a free download.

Security and Privacy

An Android application runs in its own space and cannot access data from other applications without user permission whereas iPhone applications can access many system resources by default, therefore letting the applications access user information without user permission. This makes Android applications safer than iPhone

applications from an architectural viewpoint. Because users control which services an Android application can access, they control their own security and privacy.

Although Android puts the control in the users' hands, users aren't totally protected from malware. Some applications might need to access specific services such as contacts, network etc. A legitimate application will use these services only for the intended purposes, but malware could use them to communicate personal information to a website.

Since the Android Market is open, people can easily create and market Trojan horse applications. Google removes applications from the Android Market that are identified as malware based on user experience, which means harm could be done before a malware is removed. This is a potential issue for the Android market, which could be overcome by the introduction of anti-virus software.

3.3.2 Application Development Methodology

The design methodology laid out so far is mostly based on potential user involvement in all the stages of the development. This section discusses the development approach undertaken in developing GreenAPP and the pivotal role users play in it through the use of prototyping and RAD.

Interactive System Design background

Although GreenAPP is a mobile application, it is an interactive system and can benefit from the use of HCI guidelines. According to Dix et al. (2004) HCI has three elements; the human, the computer and the interaction. Designing a successful interactive system constitutes taking into account factors related to each of those three elements.

Firstly, the human's or user's capabilities and restrictions as well as their background should be considered when designing interactive systems. Vision, hearing, touch and movement are the human tools used to interact with computing devices. The user reaction to an application they use depends on his/her senses perception of that application. Dix ibid demonstrates that other important factors in the interaction process are memory (short term and long term), reasoning, problem solving, skill acquisition, emotion, mental models and individual differences. These factors alongside human input/output tools and other psychological factors have been used to derive

interaction design rules. Secondly, advances in computing technologies such as increased processing and storage capabilities, smart phones, touch screens and wireless sensors are widening the horizons of designers and allowing them to be creative more than ever before. Finally, in order for the designer to understand interaction, the interaction process is modelled using frameworks and models such as Norman's execution-evaluation cycle which is considered by Dix as the most influential model in HCI.

Usability

In order to achieve usability, designers use good design practice paradigms to produce their applications (Dix et al., 2004). One example is the use of multimodality (using more than one human communication channel or sense to achieve interaction). Another example is the use of sensors and context aware interaction.

Interaction design Approaches and rules

Saffer (2009) introduces four types of interaction design; User Centred Design (UCD), activity centred design, system design and genius or rapid expert design. Saffer acknowledges that UCD is the most effective approach. However, what he calls genius or rapid expert design is the most dominant, especially when competition is involved.

Dix introduces the more traditional approach to design such as the iterative waterfall model and states that HCI experts and usability engineers need to be involved at each step of the design process, especially the front end design.

It is widely argued that involving the user at each step of the design is essential but very impractical when following the traditional waterfall model. Using the "Agile" approach (a working prototype of the application is produced and modified based on users' feedback) to developing application and products is focused around the users and their experience of the system. Cockburn & Highsmith (2001) discuss the positive effect of using Agile on the development process.

Designing an interactive system is bound by standards derived by the underlying theory. Sets of guidelines and heuristics are also introduced to promote good design practice. Dix mentions some examples of those rules and heuristics such as Shneiderman's eight

golden rules of interface design, Norman's seven principles for transforming difficult tasks into simple ones and Nelsons' heuristics.

Wasserman (2010) conducted a survey amongst developers on their chosen methodologies for mobile application development. The survey showed that the methodologies followed by developers are highly dependent on the mobile platform they are developing for and the toolkits provided by the vendors. He relates the results to the nature of the application being developed; that is, applications are relatively small.

In many respects, developing mobile applications is similar to software engineering for other embedded applications. Common issues include security, performance, reliability, and storage limitations. However, mobile applications have some additional unique requirement such as potential interaction with other applications, sensor handling and limited battery life.

The chosen method of development for building GreenAPP is the Agile based prototyping approach or the Rapid Application Development "RAD" which is a software development process that involves iterative development, the construction of prototypes, and the use of Computer-aided software engineering (CASE) tools. Using RAD integrates nicely with the Android SDK since Android uses an object-oriented structure approach to development and RAD encourages the same approach. Prototyping was used while designing and implementing GreenAPP. Changes to the design were carried out in consultation with users.

User Interface Design

Fling (2009) lays out the elements of a successful mobile interface; design, Context, Message, Look and feel, Layout, Colour, Typography and Graphics. Resolution and density independence is a very important aspect to take into account when designing the user interface for an android application; this is due to the vast number of devices running Android with a variety of screen sizes and pixel densities. The design of the GreenAPP followed some of the best practices for ensuring resolution and density independence through minimal use of graphs and the reliance on features available in the Android SDK.

3.4 Summary and Conclusion

A road map of the methodology and design issues was laid out. The research methods are designed to support every step of the application development process; the survey supports the design of the application as well as answering the research question, the interviews complement and expand on the results of the survey as well as getting users' views on the application, the trials and observations test the application while answering the research question. Overall the methodology deploys different research methods to converge, verify, correspond and triangulate the findings.

Chapter 4 Product Development

This chapter describes the implementation of the GreenAPP application alongside the implementation of the data transmitter

4.1 Data Transmitter Driver Software Implementation

Prior to using the embedded device data transmitter described in section 3.2.2 a Java based computer application or driver was developed to receive data via a computer's serial port, parse it and send it to the remote server. The Arduino based and the computer based transmitter software use the same methodology. This section will describe both.

4.1.1 Computer- based data transmitter

The programming environment is "Processing" which is an open source IDE based on Java (Processing, n.d). Figure 4-1 shows the Processing IDE as well as some the GreenAPP transmitter source code.

Figure 4-1 Processing development environment

Design

The functions within this driver are similar to the functions for the embedded device driver described in section 4.1.3. However, the computer driver employs the resources available to a computer user such as a screen and keyboard. An additional GUI informs the user of the current status and readings as well as plotting the data in graphical format. It also allows the user to decide on the transmitting frequency and remote server security data. Figure 4-2 shows the graphical user interface of the application.

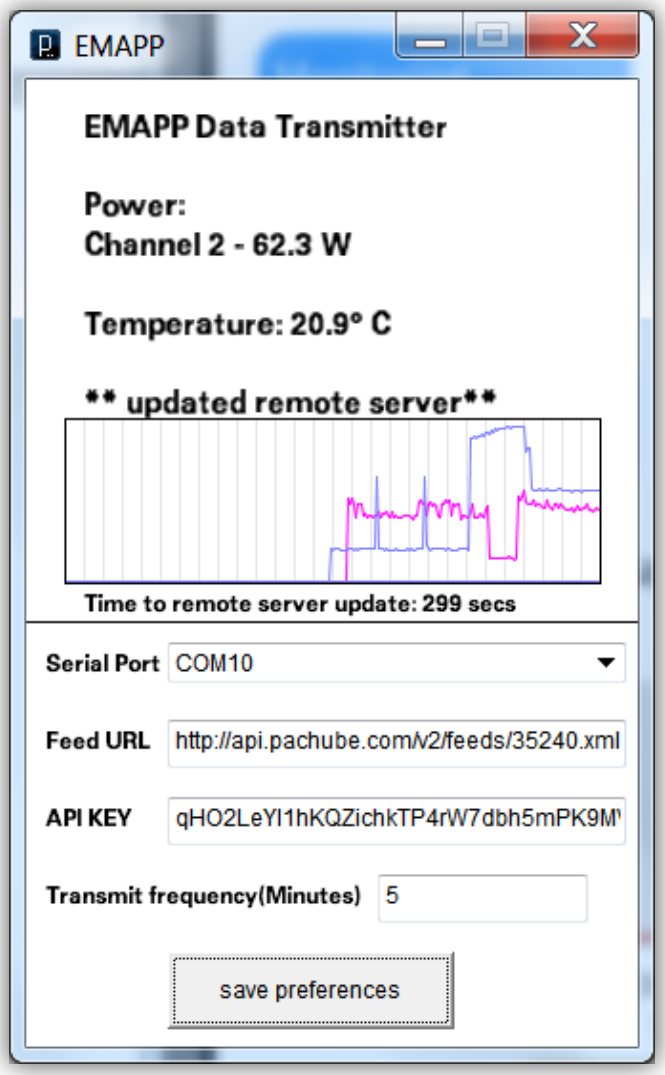


Figure 4-2 GreenAPP Data Transmitter multi-platform

Since the application is developed in Java, it can be deployed as a Linux or Windows application as well as Java Applet. The code is listed in Appendix G.

4.1.2 Arduino-based transmitter

The microcontroller used by the transmitter is programmed using the Arduino programming language (Arduino, n.d.). Although Arduino is a C/C++ based language, it has some similarities with Java programming language.

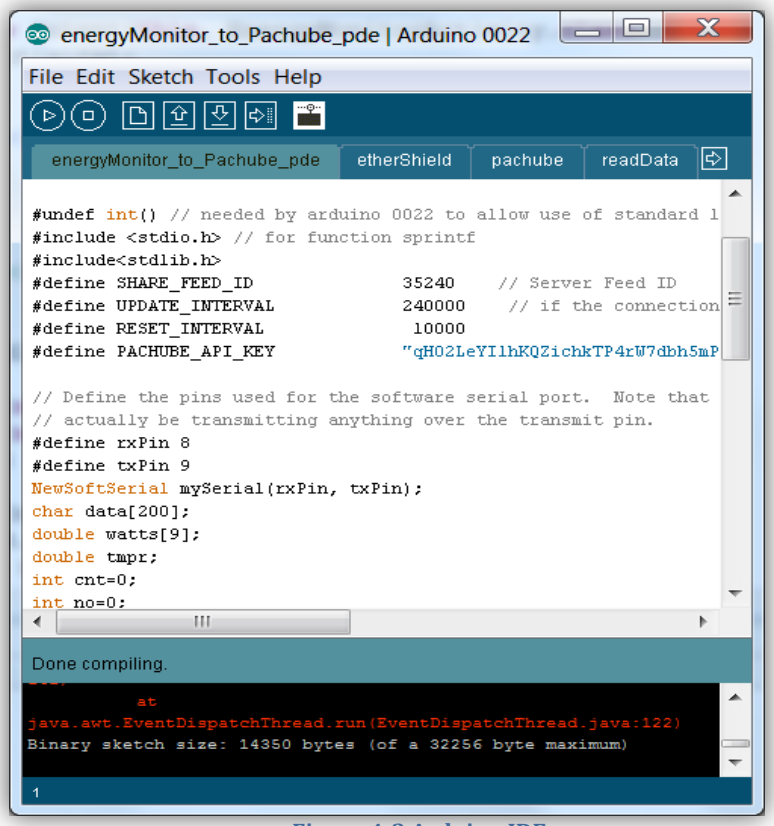


Figure 4-3 Arduino IDE

4.1.3 Design and Implementation

An Arduino program has three main components; structures, variables and functions. There is a specific manner in which an Arduino program operates, due to the fact that it is used to control or operate the microcontroller and hence the embedded device. When the system boots `setup ()` then a `loop ()` is executed until the device powers off. This requires initiations and connection setups to be placed in the setup section and any other operations that need to be repeatedly executed to be placed in the loop section. Connections to serial ports as well as Ethernet connections are controlled via functions defined in libraries provided in the Arduino programming language. The following snippet shows the format of the data the monitoring system controller transmits.

```
<msg><src>CC128-v1.29</src><dsb>00178</dsb>
<time>23:28:47</time><tmpr>21.0</tmpr><sensor>0</sensor>
<id>04073</id><type>1</type>
<ch1><watts>00045</watts></ch1></msg>

<msg><src>CC128-v1.29</src><dsb>00178</dsb>
<time>23:28:48</time><tmpr>21.0</tmpr><sensor>1</sensor>
<id>03454</id><type>1</type>
<ch1><watts>00101</watts></ch1></msg>

<msg><src>CC128-v1.29</src><dsb>00178</dsb>
<time>23:28:53</time><tmpr>21.0</tmpr><sensor>0</sensor>
<id>04073</id><type>1</type>
<ch1><watts>00041</watts></ch1></msg>
```

Figure 4-4 Serial Data structure

This data is in XML format, each message has a root element "`<msg></msg>`" which indicates the beginning and end of each message. The `tmpr` child element contains the current temperature reading in degrees centigrade while the "sensor" element indicates the appliance from which the data is being transmitted (0 indicates the main channel usually connected to the electricity meter). The `watts` child element contains the power reading in Watts.

The aim of the microcontroller's driver's program is to capture this data from the serial connection and convert it to a series of CSV values then attach it to a PUT HTTP request to the web server and then send this request through the Ethernet connection.

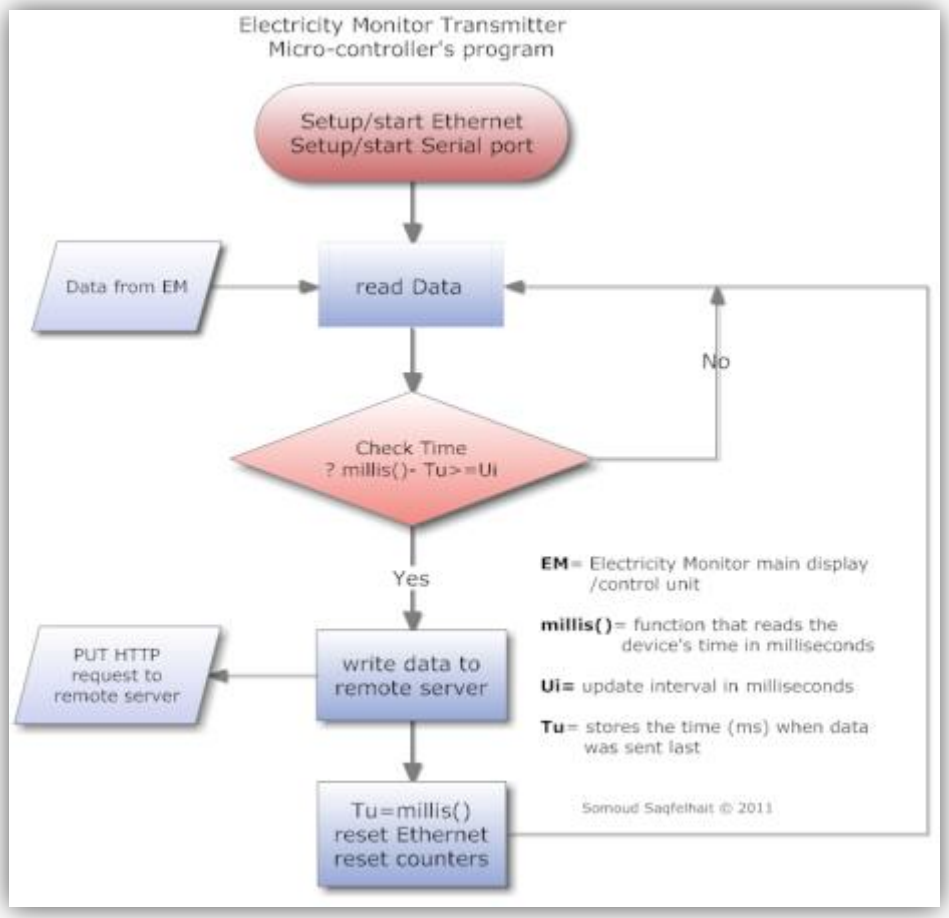


Figure 4-5 Microcontroller's program Flowchart

The flowchart in Figure 4-5 shows a diagrammatic representation of the processes within the main driver program of the microcontroller. The Ethernet connection "to the Web Server", as well as, the serial port connection "from the Monitor control unit" are initialised in the setup processes. The next step is to attempt reading data from the serial port. Then the data is either encapsulated in a PUT HTTP request using String operations and sent to the web-server if enough time has passed since the last data was sent based on a pre-set update interval, otherwise, the data continues to be read and stored in the data collectors. The following figure shows the format of the HTTP request sent to the web-server.

```
PUT /api/FEED_ID.csv  HTTP/1.1
Host: pachube.com
X-PachubeApiKey: API_KEY
User-Agent: Arduino (Pachube In Out v1.1)
Content-Type: text/csv
Content-Length: data_length
Connection: close

16.6,2352.6,35.8,193.5 .....
```

Figure 4-6 PUT HTTP request

Since the data connection is serial the data arrives character by character.

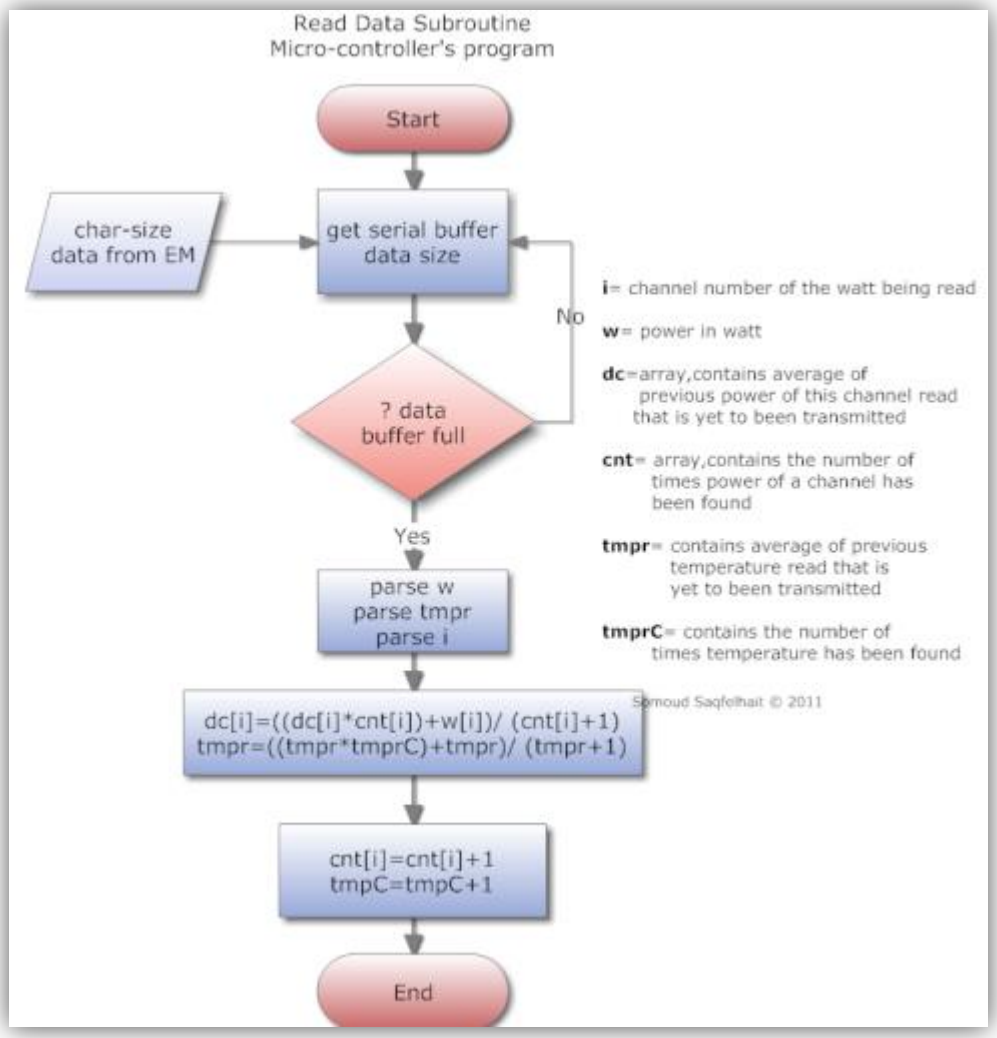


Figure 4-7 Read Data Subroutine Flowchart

Figure 4-7 shows the "read Data" subroutine. It uses an array of numbers to store the averages of the readings by also keeping track of the number of readings for each individual item. The amount of power and temperature readings the transmitter is

dealing with is massive; it needs to keep track of 11 ever changing data values. The web-server accepts data for a single feed every 10 seconds during which the data values would have changed up to 10 times. This presents a dilemma; what set of data would the program encapsulate in the HTTP request? The last reading or the averages of all readings since the previous set of data were transmitted? The latter was chosen due to its better accuracy. Even though the web server allows for updates every 10 seconds, the amount of data that will be generated is huge (9 thousand updates a day or over 3.2 million updates a year). The number of readings stored in a single request should not be very big, risking losing that information if the request was not received by the server. After many experiments the update interval was set to 4 minutes which reduces the number of updates by 96%. The overall source code is listed in Appendix H.

4.2 Energy Monitor Application Development

This section describes the implementation of GreenAPP. The section starts with a description of the development environment and the Android SDK. Some of the essential structures and files are illustrated

4.2.1 Development Environment

Android applications was developed using the Android Software Development Kit "SDK" and the Android Development Tool "ADT" plugin for Eclipse IDE.

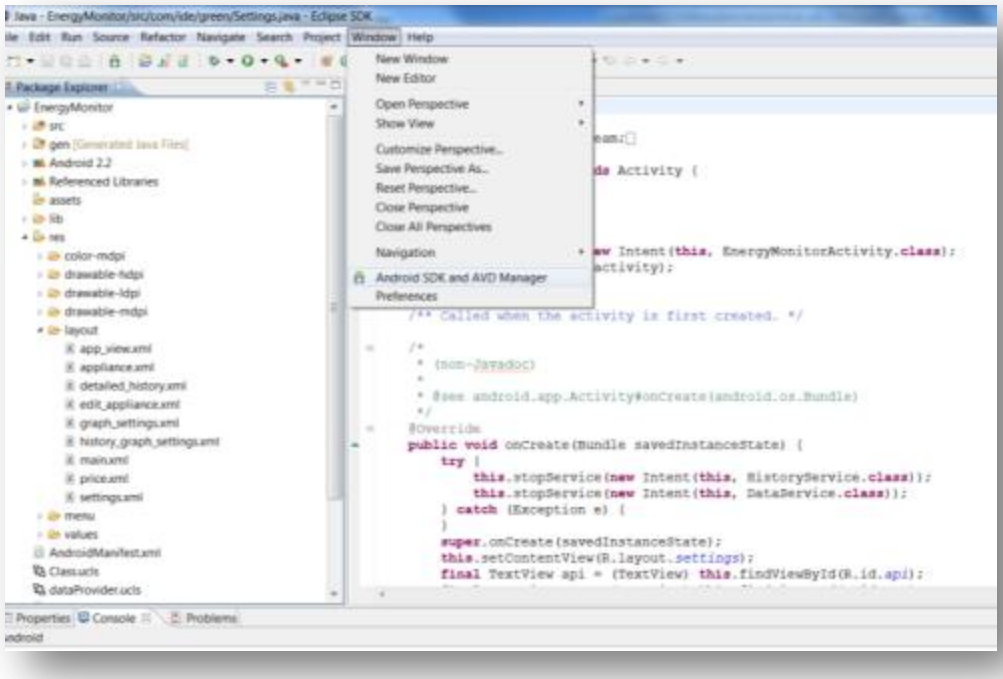


Figure 4-8 Eclipse and file structure of an

4.2.2 Application Structure

The application structure of any Android application was illustrated in Figure 3-8 under Android platform. It includes activities, services, content providers, etc. The AndroidManifest.xml contains a listing of the structural elements of any android application.

AndroidManifest.xml

The most important `.xml` file is the `AndroidManifest.xml` file because it delivers the essential details about the application to the Android system without which the system cannot run any of the application's code (Android, n.d.).These details are

- The application package which acts as a unique identifier for the application; GreenAPP's package is `com.ide.green`.
- The components of the application; activities, services, broadcast receivers, and content providers.
- The processes that will host application components.
- The permissions the application must have in order to access protected parts of the API and the permissions other applications are required to have in order to interact with the application's components. GreenAPP requires a full internet access permission in order to be able to request data from remote platform.
- The minimum level of the Android API that the application requires.

Figure 4-9 and Figure 4-10 illustrates the information held in GreenAPP

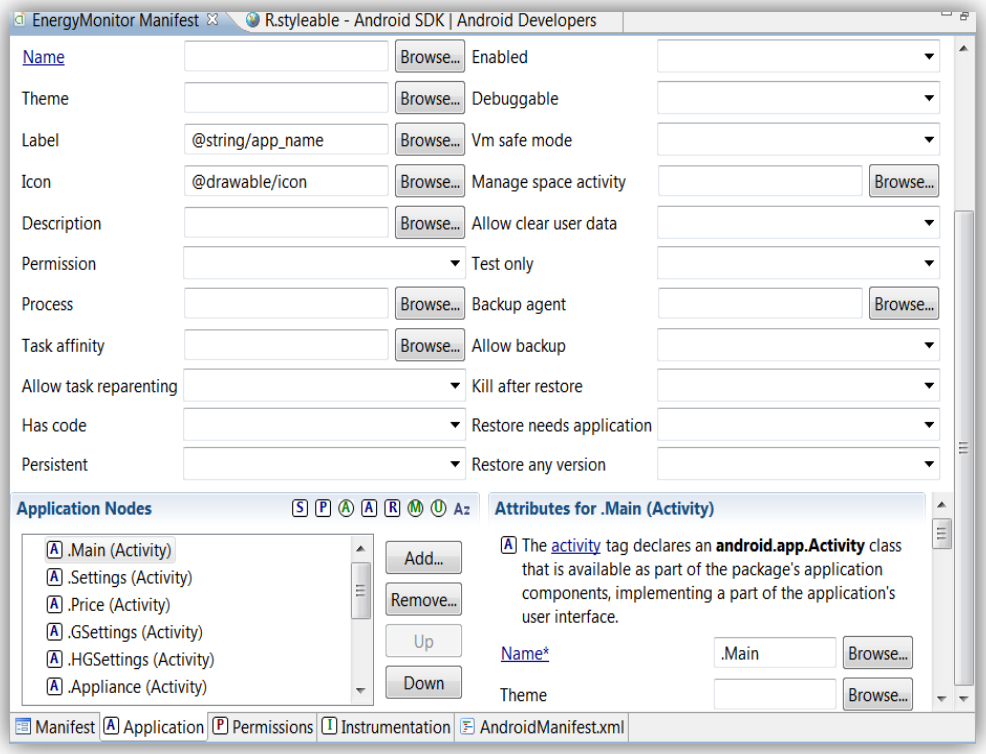


Figure 4-9 AndroidManifest.xml of GreenAPP graphical view

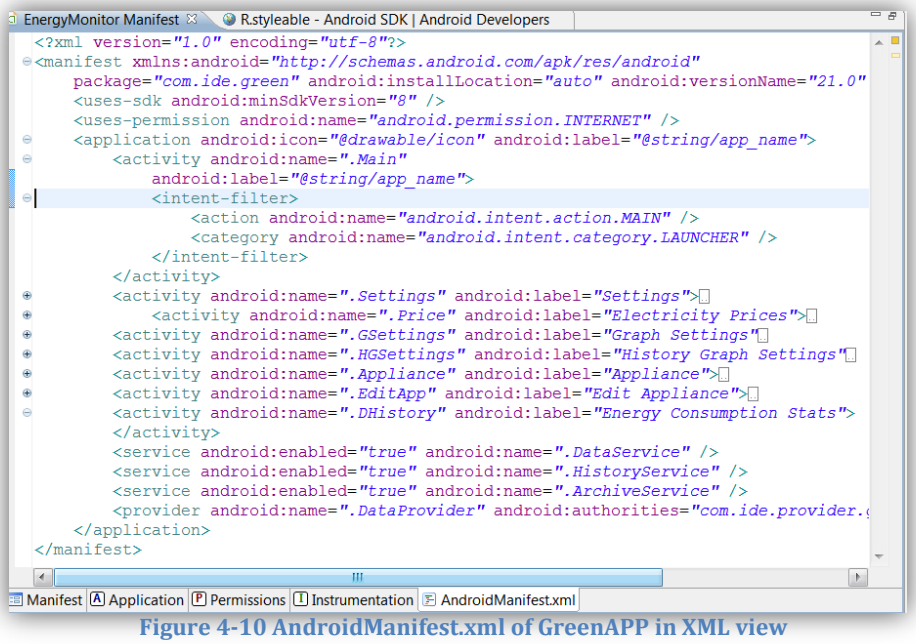


Figure 4-10 AndroidManifest.xml of GreenAPP in XML view

4.2.3 User Interface Implementation

In Android the user interface is implemented using View Objects; which are objects whereby the layout attributes and content for a specific rectangular area of the screen is stored. It is also a point of interaction for the user with the application. The layout of any screen or activity in an Android application is defined in `.xml` file in the resources area of the project.

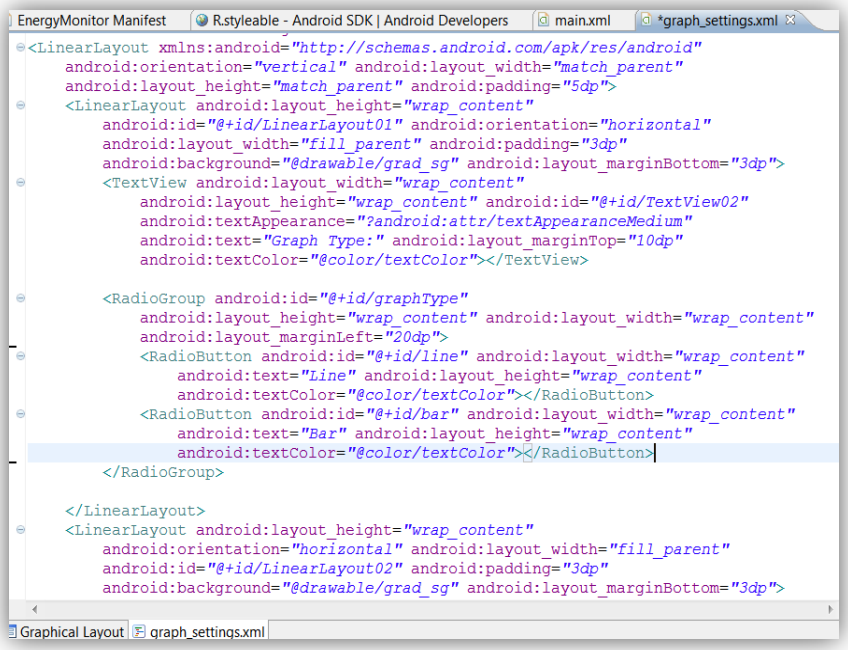


Figure 4-11 XML layout of graphSettings layout

Figure 4-12 Graphical layout of the

Figure 4-11and Figure 4-12 show an example of a GreenAPP Activity View design in XML and graphical layouts respectively. Not only is the resources area used for the user interface layout but it is also used for storing colours, strings, numbers, arrays of strings and numbers which separate the user interface design attributes from the functionality or the programming part. An example is shown in Figure 4-13. Notice that the string array "range" is the same array used to define the spinner's entries `@array/range` in the properties part of Figure 4-12

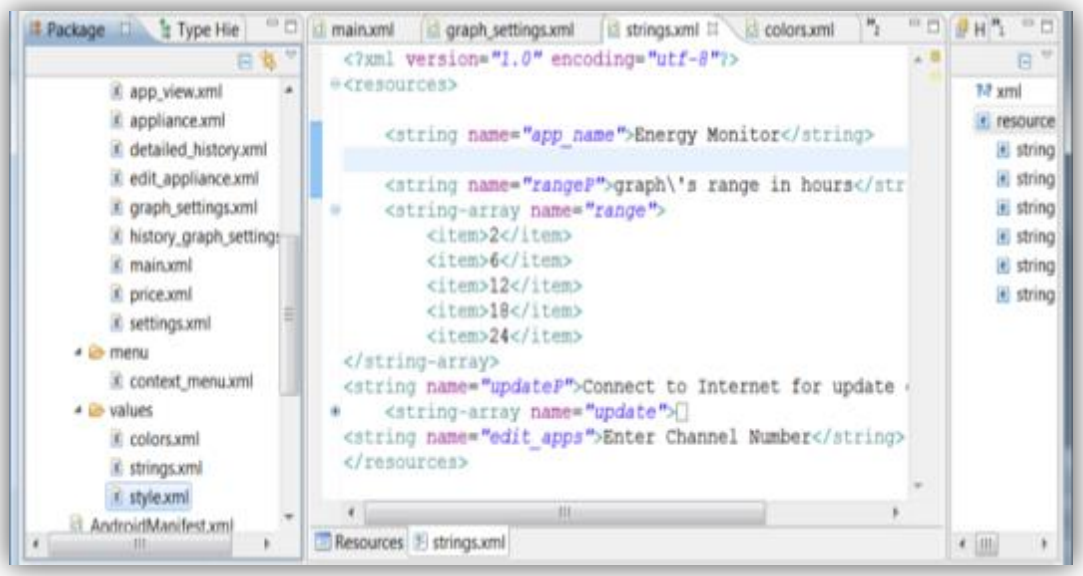


Figure 4-13 XML layout of graphSettings layout

4.2.4 Functionality Implementation

The implementation of an android application requires the use of specific android components which are the crucial building blocks of the application's operation; as they act as entry points to the application either for the end user or the system itself. These components are: Activities, Services, Content Providers and Broadcast Receivers (Android Developer, n.d.). Activities are associated with the user interface UI, each activity corresponds to a single UI or a screen, Figure 4-14 shows the main Activity of the EnergyMonitor mobile application GreenAPP.

Services allow applications to perform complex calculations, connect to a database connect to a Web Service or even play music in the background without interacting with the user. The use of Services has been the foundation that enabled GreenAPP to display real time data and a dynamic graph. A Service is initiated and terminated by an Activity through the `StartService` and `StopService` functions respectively. A Service or an Activity usually interacts with other Services or Activities through Broadcast Receivers. Broadcast Receivers respond to broadcast announcements which are system wide messages also known as Intents containing either data or action information. A Content Provider offers an organised and easy way for Activities and Services within an application to interact with databases and files. GreenAPP implements a content provider to interact with a SQLite database whereby all the energy monitoring data received is stored permanently for future use. The use case diagram in Figure 4-15

draws a general picture of the functionality provided by the GreenAPP system.

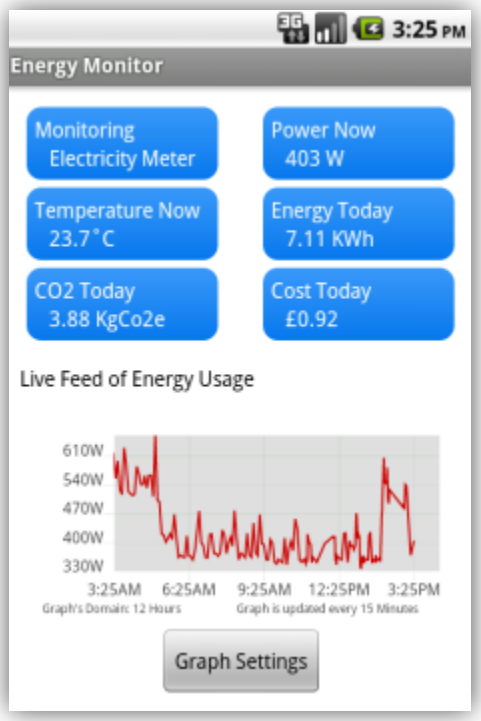


Figure 4-14 Main Activity of GreenAPP

In GreenAPP three Service classes are implemented to connect to the remote web server and read three types of data; the `DataService` gets the current energy readings, the `HistoryService` deals with readings since midnight and the `ArchiveService` reads data taken in the last 3 months from the remote web server. Each of these services relies on an implementation of Content Provider to access two data tables. The first table "data" stores data acquired by the `DataService` and the `HistoryService`. This table is used by the main activity to create the dynamic graph. The second table "history" stores data acquired by the `ArchiveService` and is used by the `detailedHistory` activity to draw a statistical summary graph as shown in Figure 4-16.

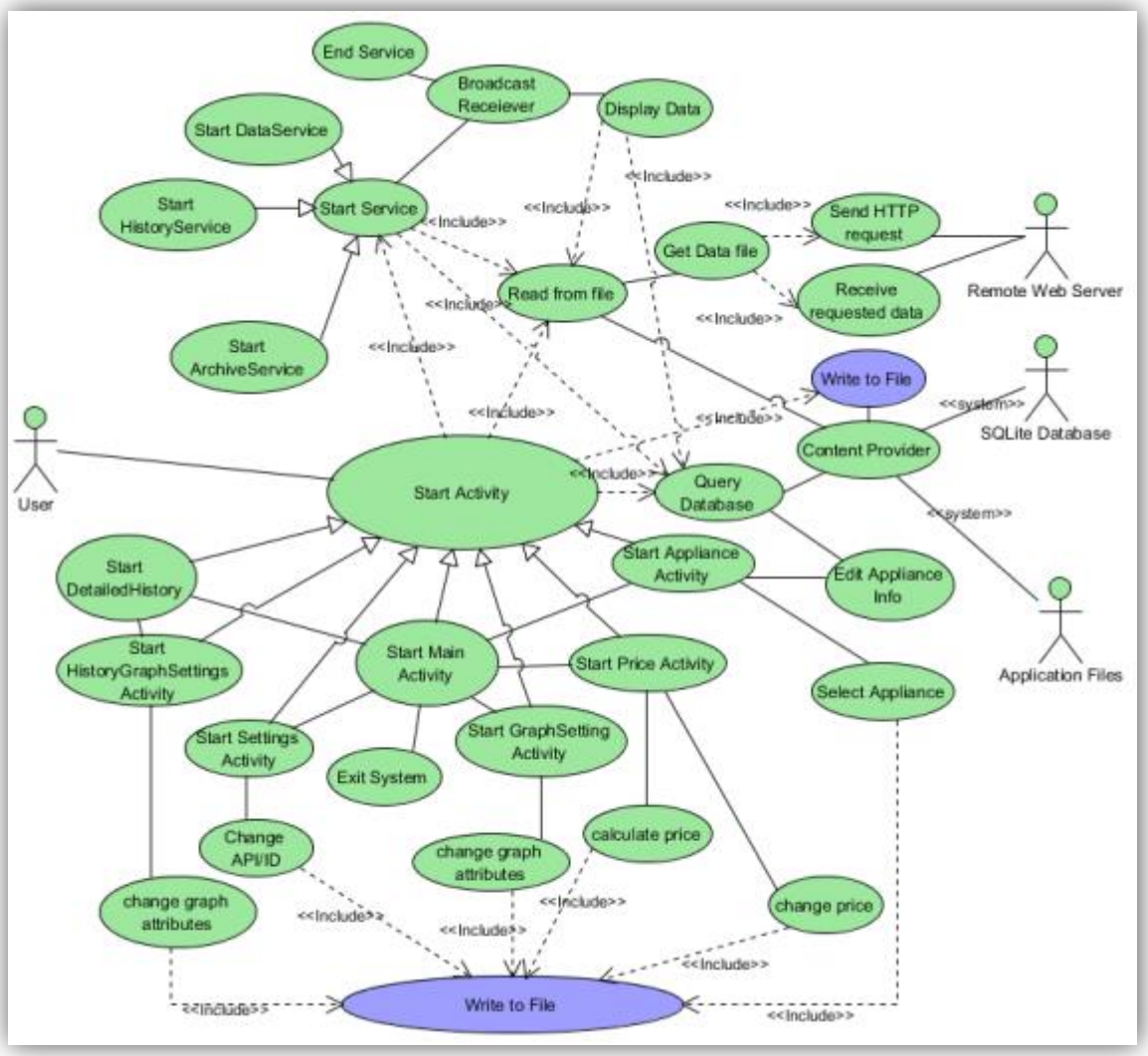


Figure 4-15 Use Case Diagram of the system

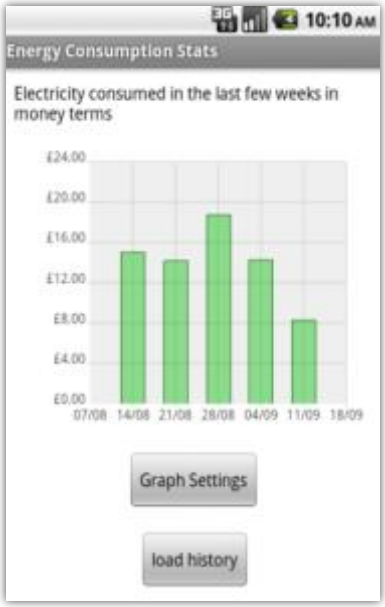


Figure 4-16 DetailedHistory Activity

The ERD in Figure 4-17 shows the tables in the system database and their attributes.

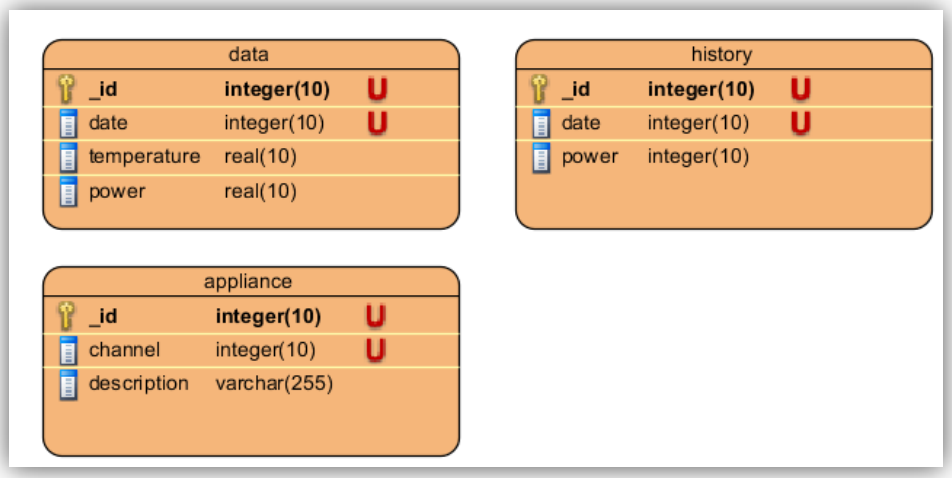


Figure 4-17 GreenAPP database Entity Relationship

The sequence diagram in Figure 4-18 shows the GreenAPP system activity objects interaction in a time sequence starting with the `Main` Activity which is the main entry point to the system.

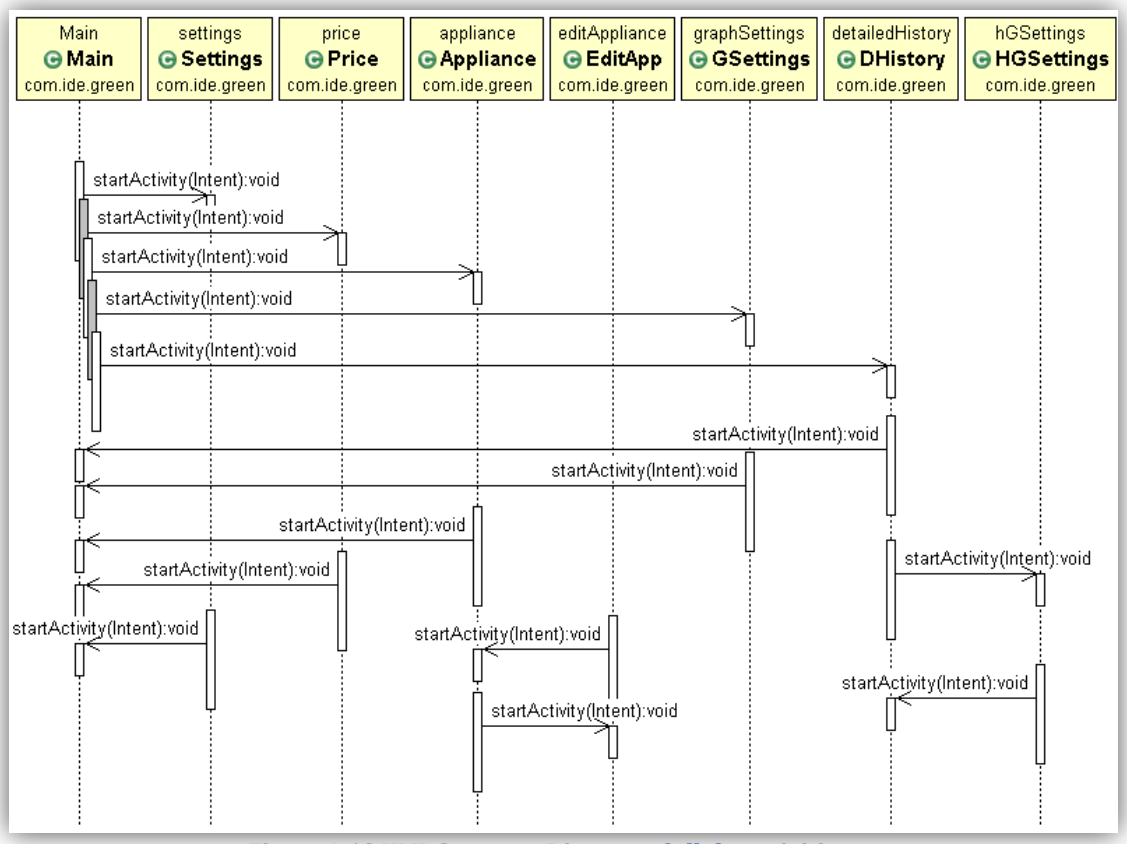


Figure 4-18 UML Sequence Diagram of all the activities

The user can navigate from the main screen to most of the activities using a context menu as shown in Figure 4-19.

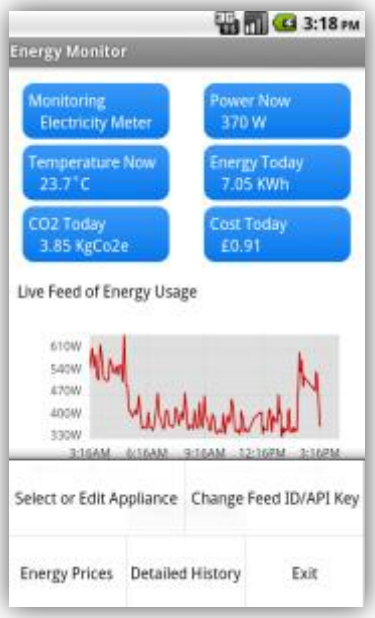


Figure 4-19 main navigation system of GreenAPP

The class diagram in Figure 4-20 describes the most of the structure of the GreenAPP system, only some of the operations are shown for the parent classes.

The sequence diagrams for the system's services and their interaction with the system's activities are shown in Figure 4-23 for the main activity interaction with `HistoryService` and `DataService` and in Figure 4-24 for the detailed history activity and the `ArchiveService`. Figure 4-21 illustrates the class diagram of the three services and the classes they use. The three services are designed in a similar way with slightly different functionalities. They use an implementation of the `AsyncTask` class to read the data from the remote web server and store in the SQLITE database with the help of the `DataProvider`; an implementation of `ContentProvider` class. The `AsyncTask` class offers a simple, convenient mechanism for moving time-consuming operations, such as connecting to remote web server, onto a background thread. It handles the thread creation, management, and synchronization, facilitating the creation of an asynchronous task consisting of processing to be done in the background and a UI update to be performed when processing is complete.

The implementation of the `DataProvider` class is represented in the class diagram in Figure 4-22. It uses an implementation of the `SQLiteOpenHelper` class i.e. `dataDatabaseHelper` class, to facilitate and organise access to the SQLITE database.

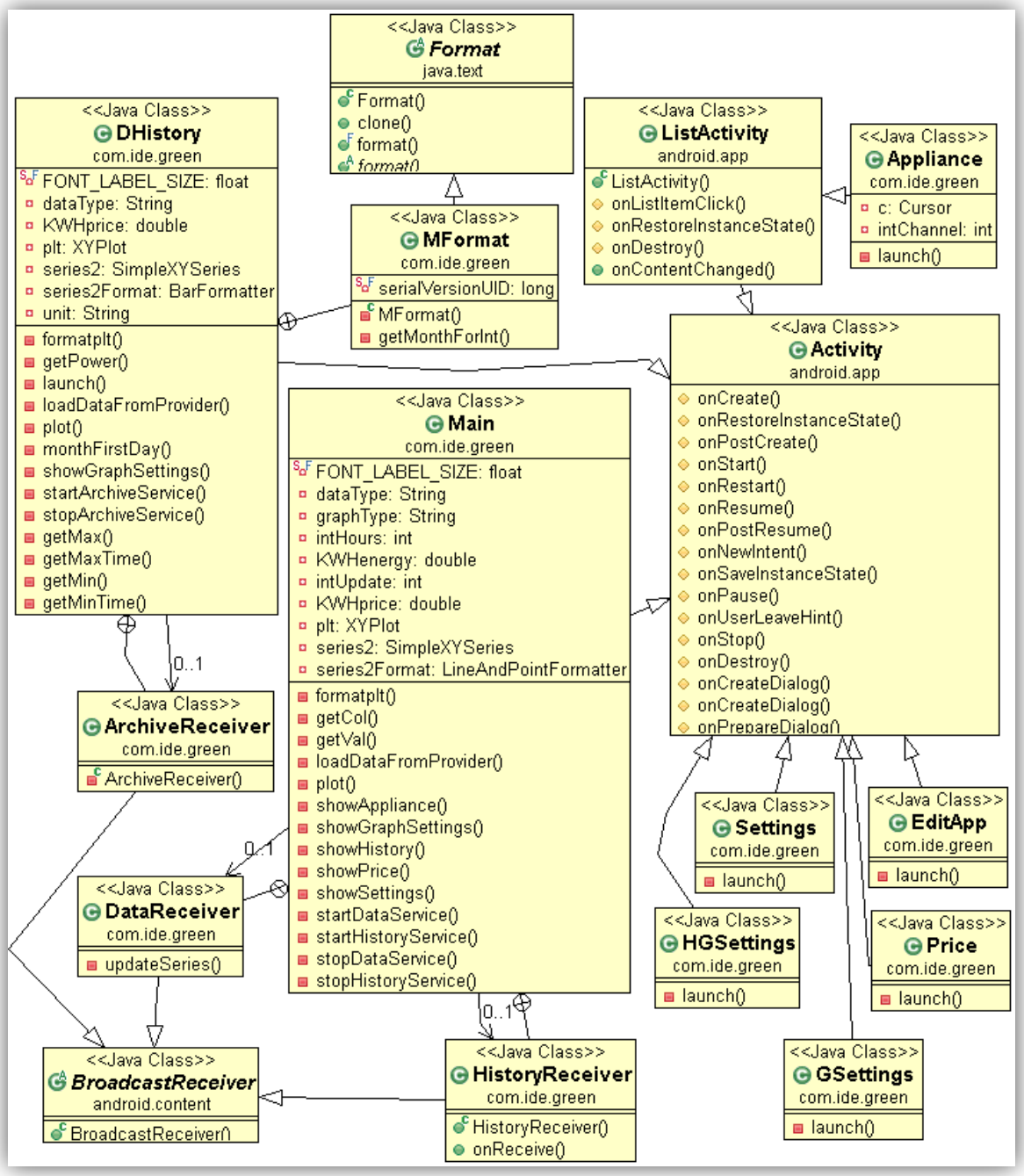


Figure 4-20 UML Class diagram of all Activities

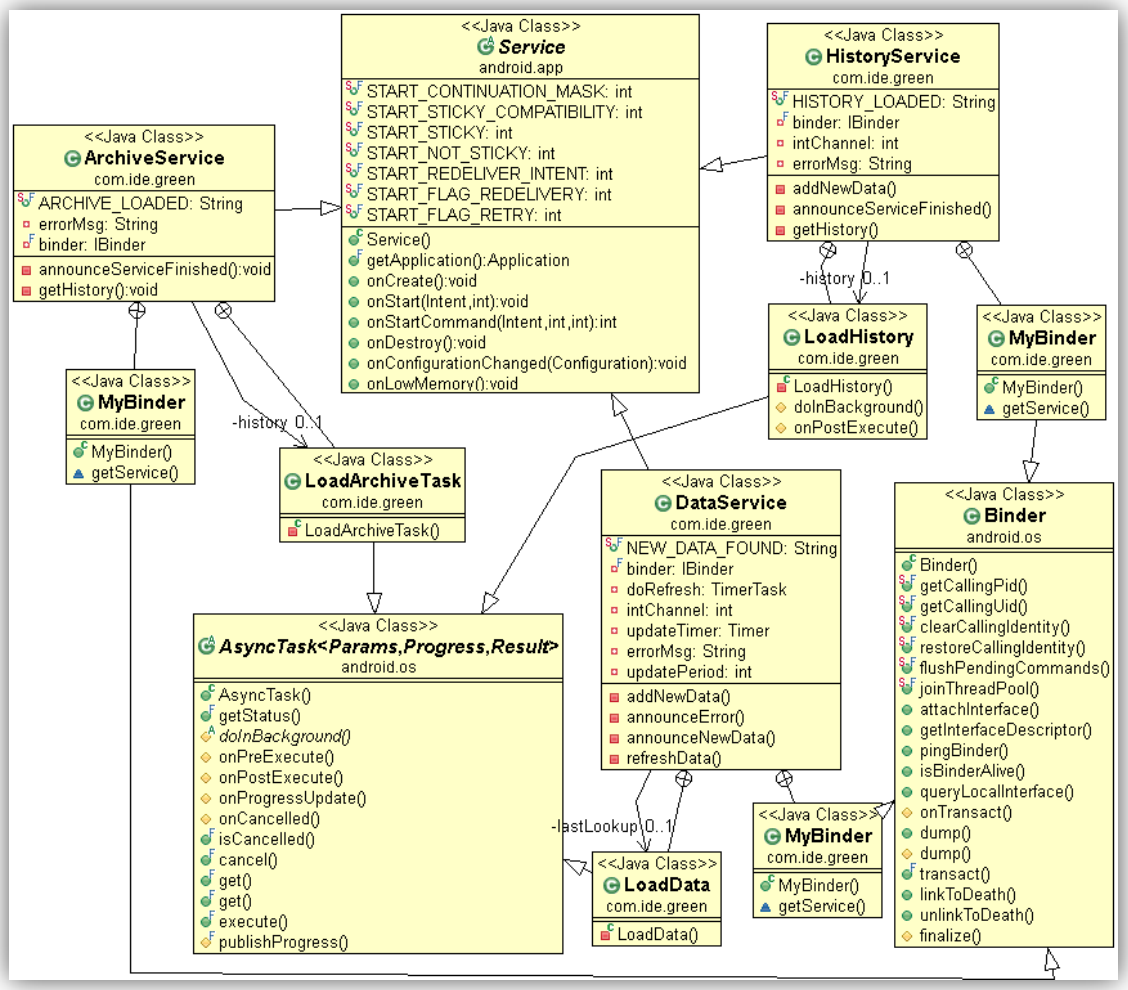


Figure 4-21 class diagram of GreenAPP services

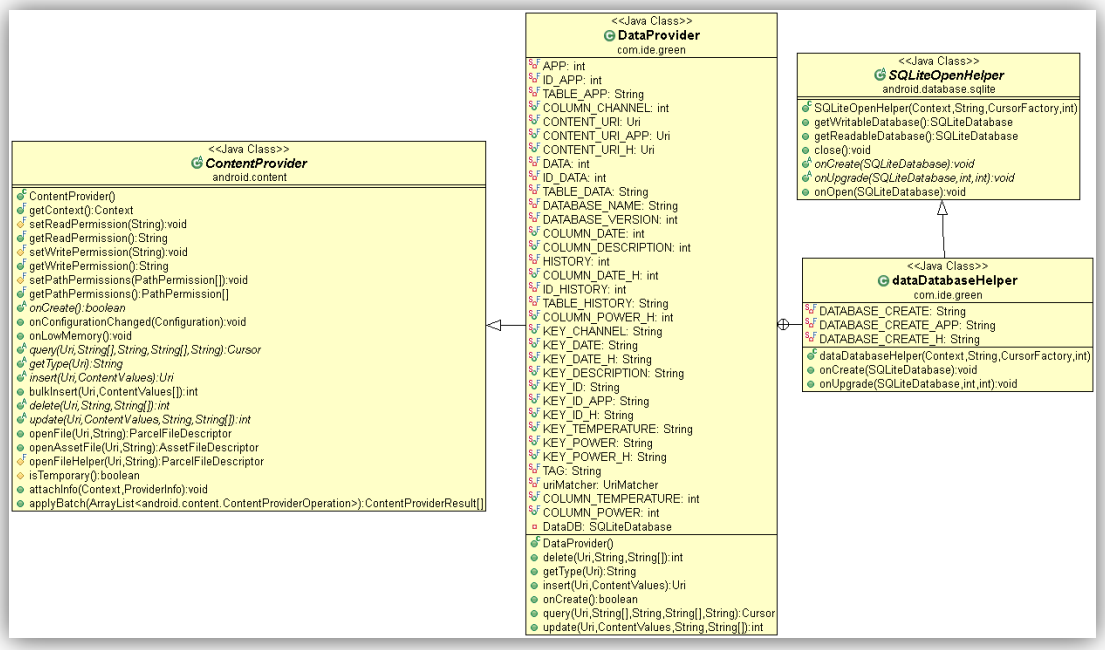


Figure 4-22 Implementation of the GreenAPP DataProvider

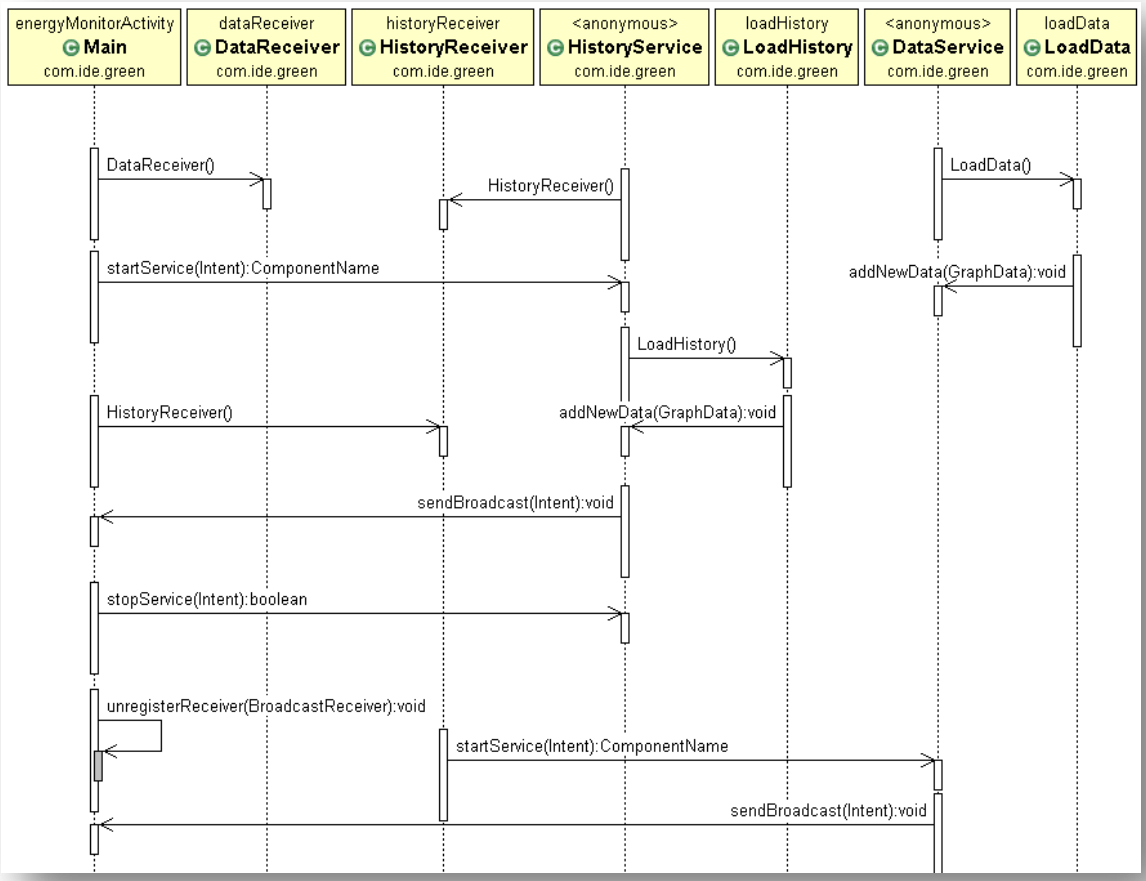


Figure 4-23 Sequence Diagram of Main Activity Interaction with Services

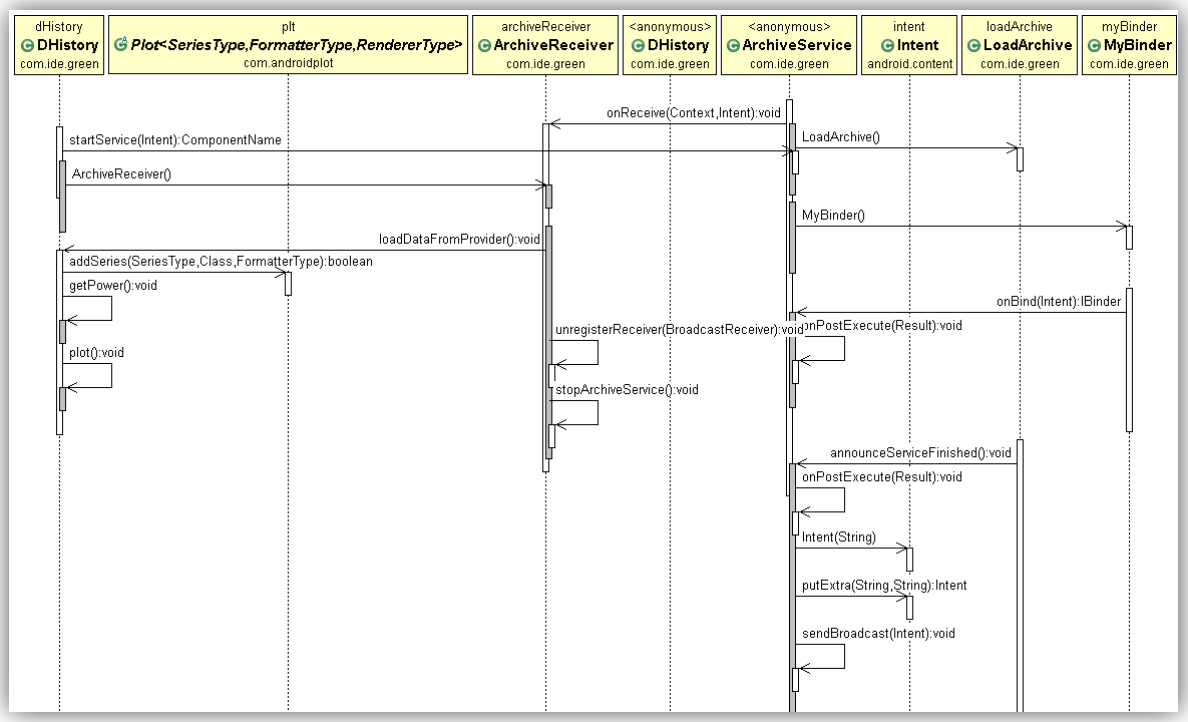


Figure 4-24 sequence diagram of the relationship between detailed history activity and ArchiveService

Dynamic Plot

An external library "AndroidPlot" was utilised to implement the graph facilities. When activity is started, the SQLite database is checked to see if there is any historical data available. If no data is available the `HistoryService` is launched to get the data from the remote server. Upon finishing its task the `HistoryService` send a broadcast which gets received by the `historyReceiver` which in turn calls the plot functions to plot the data retrieved. The `historyReceiver` then launches the DataService operate by initiating a timer and scheduling an `AsyncTask` to check the remote server for new data in a frequency based on the settings entered by the user. The check is done by reading the current data file from the server, parsing it and comparing the data's time stamp with the ones stored in the database. Upon finding new data, the `DataService` send a broadcast with the data found attached as extras. The broadcast get received by the `dataReceiver` which adds the data to the plot series and calls the plot function to redraw the graph and readjust the axes. Figure 4-25 demonstrates this process.

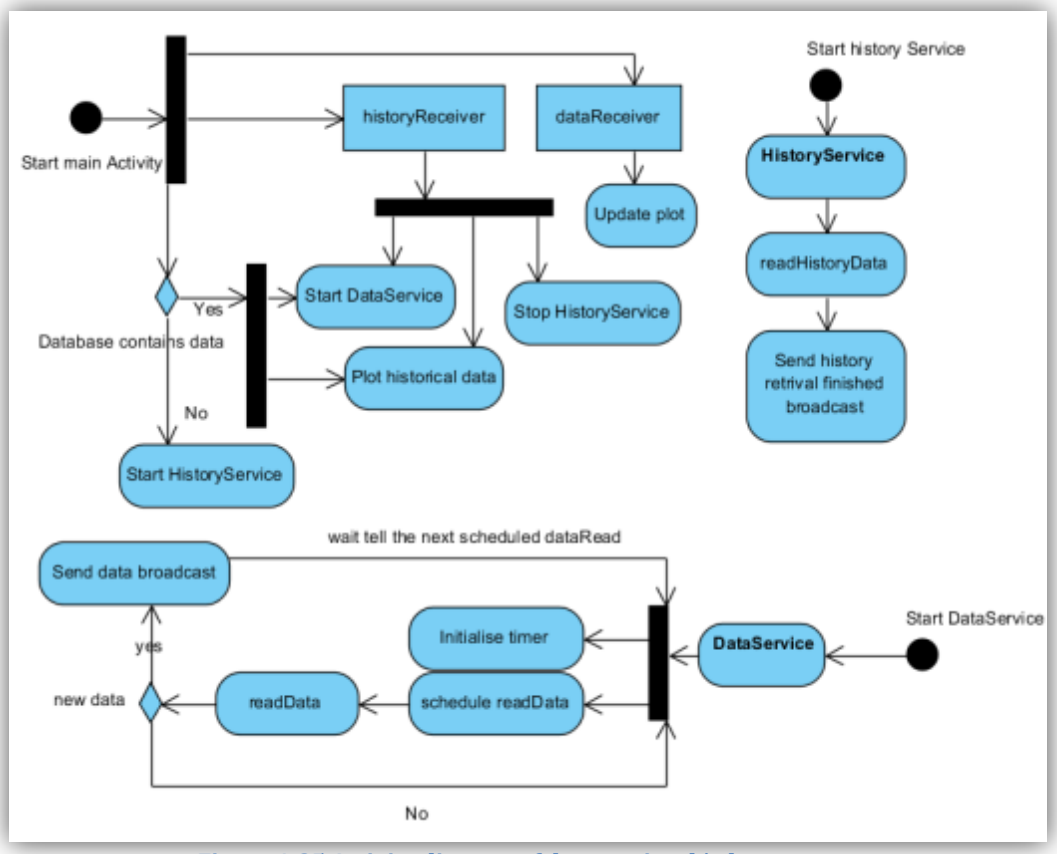


Figure 4-25 Activity diagram of data retrieval/ plot process

Since the graph is dynamic the labelling of the x-axis (domain) and the y-axis (range) proved to be challenging. Users pointed out the difficulty with which they could understand the graph. To solve this problem an algorithm was devised to set the interval between the tick marks to multiples of ten for the range.

```
// calculate the max and min of the y value to format the y axis
double max = Commons.getMax(intHours, series2, 2);
double min = Commons.getMin(intHours, series2, 2);
plt.setRangeBoundaries(min, max, BoundaryMode.AUTO);
if (max > 0) {
    double step = (max - min) / 5.0;
    int nearest = (int) Math.round(Math.log10(max - min) / 2);
    double rounded = Commons.roundToNextTen(step, nearest);
    if (rounded < 1)rounded = 1; //make sure the step is not zero
    plt.setRangeStep(XYStepMode.INCREMENT_BY_VAL, rounded);
    plt.setTicksPerDomainLabel(4);
}

plt.setDomainStep(XYStepMode.INCREMENT_BY_VAL, intHours * 60 * 60
        * 1000 / 8);
```

Figure 4-26 adjusting axes' label values the graph in GreenAPP code snippet

Parts of the source code are available in 0 and the overall source code can be found the DVD included.

Calculations

The requirements of the GreenAPP system indicate a need for some mathematical calculations. The main screen offers a display of the current power readings in a "time versus power" graphical format as well as the amount of energy consumed so far on that specific day alongside its carbon footprint and its price. This require the system to perform the following calculations

- Energy Consumption in kWh
- Price in Sterling Pounds(£)
- Carbon Dioxide Emission or Carbon footprint in $kgCO_2e$

The following paragraphs illustrates how these calculations are performed and how the methods were derived

Energy Consumption Calculations

According to GCSE physics, $Energy(kWh) = power(kW) \times time(h)$

Since the remote web server stores discrete real time values of power and time, the energy consumed by an appliance is the area under the curve of the Power versus Time Graph (Grondzik et al., 2010)

According to the trapezoidal rule (Pidgeon& Ong, 1996), the area under the entire curve can be measured by summing the areas under each of its segments

$$AreaUnderTheCurve = \sum_{i=1}^{n} (\Delta t_i) \left(\frac{p_i + p_{i+1} + 1}{2} \right)$$

$$Where \quad \Delta t_i = t_{i+1} - t_i$$

In this case p_i is the instant reading of power in Watt at a specific time t_i and n is the number of readings for the required period. Since the data stored in GreenAPP is in Watt (W) and milliseconds (ms) the Watts have to be converted into kilo-Watt (kW) and the milliseconds into hours using

$$Power(kW) = \frac{Power(W)}{1000} \ \& \ Time(h) = \frac{Time(ms)}{1000 \times 60 \times 60}$$

The code snippet in Figure 4-27 shows the implementation of power computation for GreenAPP

Carbon Dioxide Emission/ Carbon Footprint Calculations

The carbon footprint is a measurement of all greenhouse gases produced by a certain individual or group and it has units of tonnes (or kg) of carbon dioxide equivalent. Greenhouse gas conversion factors are used to calculate the amount of greenhouse gas emissions caused by energy use. They are measured in units of kg carbon dioxide equivalent. In order to convert the energy consumed in "kilo-Watt hour" or "kWh" to "kilo-gram of carbon dioxide equivalent" or "kgCO_2e", the energy use should be multiplied by a conversion factor. The conversion table published by the Carbon Trust (2011) states that the conversion factor for grid electricity is 0.54522. So, the following formula can be used to calculate carbon emission in kgCO_2e

$$Carbon\ Emission(kgCO_2e) = 0.54522 \times Energy(kWh)$$

```
/**
 * @param power
 * @param time
 * @return
 * @throws IndexOutOfBoundsException
 */
public static double computeKWH(Number[] power, Number[] time)
        throws IndexOutOfBoundsException {
    //initialise the sum
    double sum = 0;

    Number deltaTime;
    Number deltaPower;

    for (int i = 0; i < power.length - 1; i++) {
        if ((time[i + 1] != null) && (time[i] != null)
                && (power[i] != null) && (power[i + 1] != null)) {

            deltaTime = time[i + 1].longValue() - time[i].longValue();
            //divide power by 1000 to convert to kW
            deltaPower = (power[i].doubleValue() + power[i + 1]
                    .doubleValue()) / 1000;
            /* time is in millisecond we divide by 1000 to convert to
             seconds, by 60 to convert to minutes and by 60 to convert to hours*/

            sum = sum + deltaTime.doubleValue() * deltaPower.doubleValue()
                    / (2.0 * 60 * 60 * 1000);
        }
    }
    return sum;
}
```

Figure 4-27 GreenAPP implementation of energy computation code

Price Calculations

GreenAPP allows the user to input the price they pay for each energy unit (kWh) they use. The price of certain energy consumption can be calculated by multiplying the price per kWh by the kWh

$$Price = Energy(kWh) \times Price/kWh$$

As illustrated in Figure 4-28 the GreenAPP system offers two options for the user to enter their energy price either directly or by entering the total cost and the total energy consumed and then calculating the price. This offers some flexibility to the user as energy suppliers vary in their billing systems. The following formula is used by GreenAPP to calculate the pence/kWh

$$Price\left(\frac{p}{kWh}\right) = \frac{Cost(£) \times 100(\frac{p}{£})}{kWh}$$

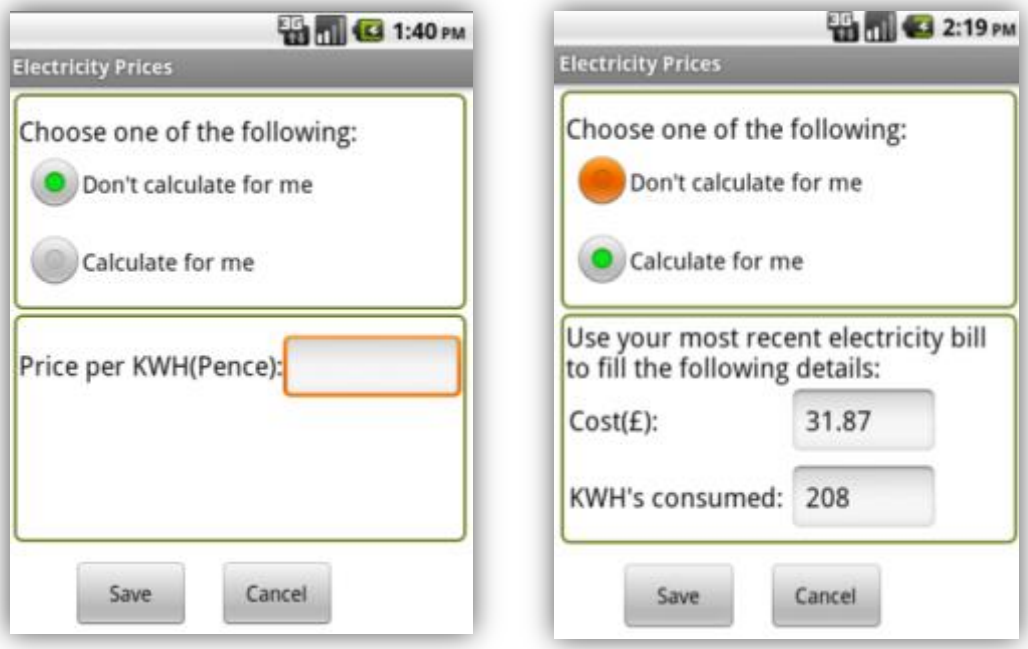
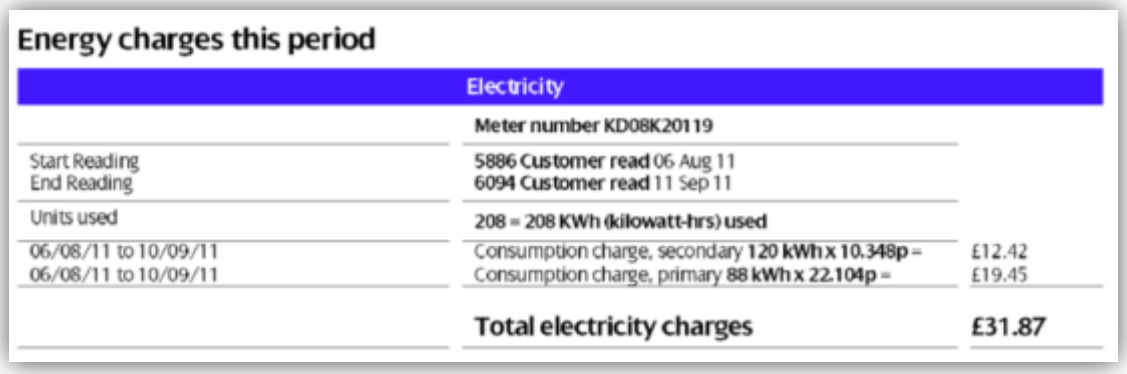


Figure 4-28 Price Activity View

For example, if a user of GreenAPP had an energy bill similar to the one in Figure 4-29, they could decide to let GreenAPP calculate their energy price since their provider charges different rates for different uses. Or they might decide to enter an average of the rates. The calculated rate by GreenAPP using the above formula is 15.3p/kWh and the average is 16.2p/KWh. Using GreenAPP calculations is more accurate since the two rates don't contribute equally to the total price.



Energy charges this period

Electricity		
	Meter number KD08K20119	
Start Reading End Reading	5886 Customer read 06 Aug 11 6094 Customer read 11 Sep 11	
Units used	208 = 208 KWh (kilowatt-hrs) used	
06/08/11 to 10/09/11 06/08/11 to 10/09/11	Consumption charge, secondary 120 kWh x 10.348p = Consumption charge, primary 88 kWh x 22.104p =	£12.42 £19.45
	Total electricity charges	**£31.87**

Figure 4-29 an extract of an energy bill

4.3 Summary

This chapter highlighted some of the implementation of the GreenAPP data transmitter and mobile application.

Chapter 5 Analysis and Findings

In this chapter, the results of the research methods used to aid the development of GreenAPP and the effect of its use will be discussed and analysed. Firstly the finding and analysis of the online survey, interviews, observations and trials will be presented. The chapter will conclude with a discussion on the effect of using these methods as part of the triangulation research technique on the validity and accuracy of the results.

5.1 Online Survey

5.1.1 Overview

An online survey aimed at computer/internet user was designed and implemented using Survey Monkey (www.surveymonkey.com) and it covered the main themes described in section 3.1.2. The ethical requirements for low risk research projects involving human participants were fulfilled by introducing the consent part of the survey and using the submit button. The survey was piloted on three individuals with different professional and personal backgrounds; this helped to identify the average amount of time needed to be spent on doing the survey to act as a guideline as well as identifying any potential ambiguities. The survey was distributed through Twitter, Facebook and email to potential participants in the Yorkshire and Humber region, particularly residents in the city of Leeds. Eighty-one people started filling the survey and seventy-four people completed it, resulting in a 91.4% completion rate. Participants were asked about their place of residence, income, views on environmental issues, incentives for energy consumption reduction, willingness to use energy monitoring devices and their preferred feedback format and design interface. The copy of the questions can be found in 0. The some of the questions are demographic or supportive. The analysis of their answers does not tell us much by itself; however they were used to analyse the rest of the questions.

5.1.2 Analysis and Findings

Most of the survey questions aim to collect measurable outcomes to inform and justify the decisions during the GreenAPP design and implementation stages. Microsoft Excel was used to analyse the quantitative data and NVIVO was used to analyse the open

ended questions. This section highlights some of the findings. More of the survey's findings can be found in 0.

Energy Saving Motivators

Plotting the response of participants to the energy saving motivators results in the pie chart shown in Figure 5-1which shows that around 54% of the participates said that their motivation to save energy is influenced equally by ecological matters and financial considerations. On the other hand 25% believed that their main motivation is financial and the remaining 20% stated that their motivation is financial. This indicates that energy consumption feedback is desired to accommodate both financial and ecological elements.

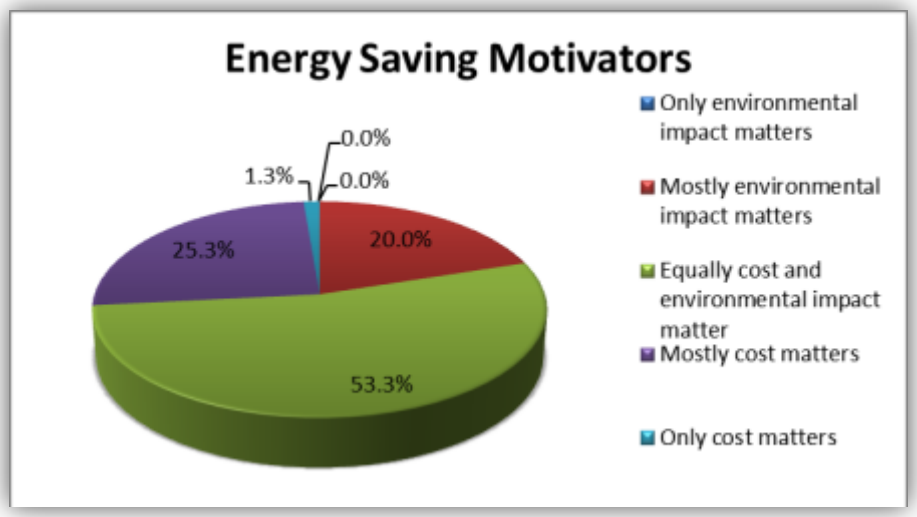


Figure 5-1 Energy saving motivators survey response

Analysing the energy saving motivators' response further produced some interesting outcomes; when comparing household income with the motivation behind energy saving (see Figure 5-2) 30% of participants in the lowest income category (10-20 thousands a year) had the environment as their most significant motivator compared to 11% for the next category(20-40 thousands a year) and almost 30% for the penultimate category(40-80 thousands a year) and for the higher income category(80 thousands and over) 25% of the participants had cost as their strongest motivator.

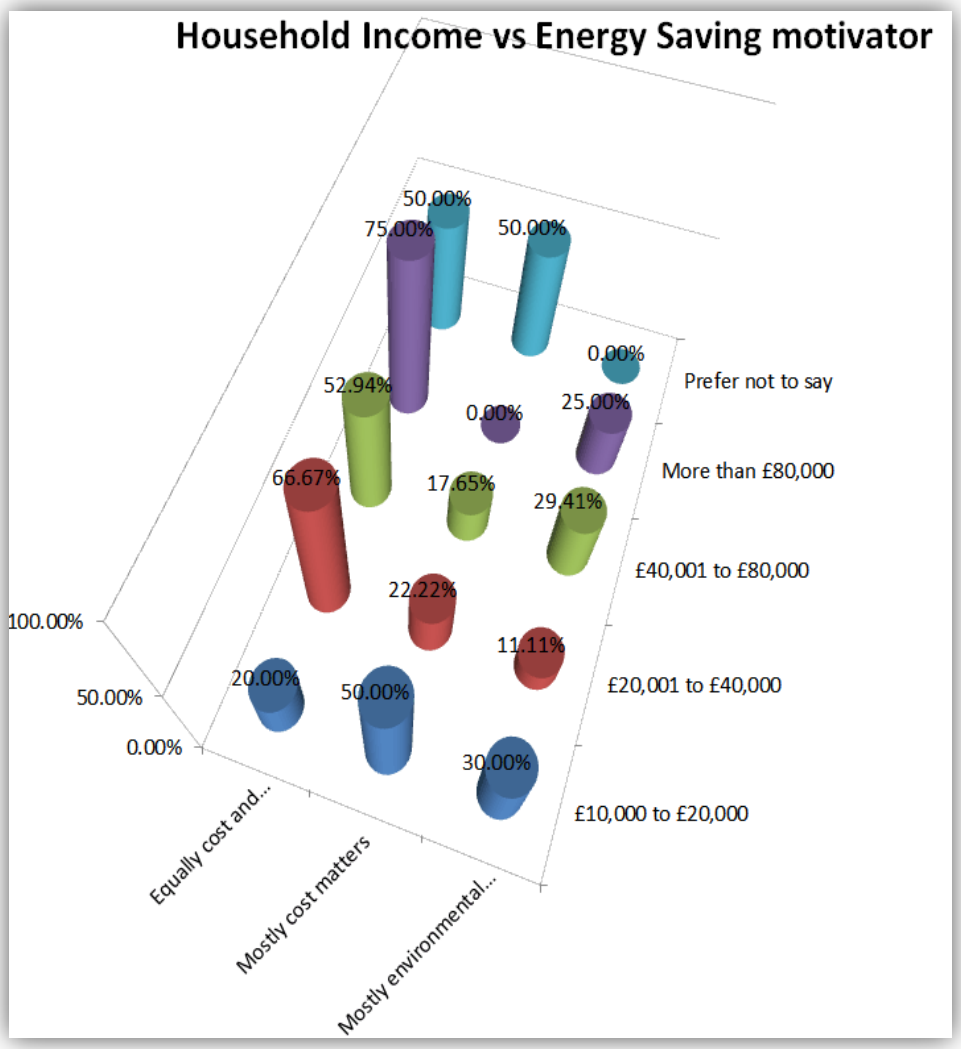


Figure 5-2 Household income vs. Energy Saving Motivators survey responses

Preferred energy consumption feedback

Comparing current energy use with historical use and receiving alerts when electricity use exceeds a specific target were the most popular types of feedback amongst 80% of the respondents. On the other hand, social media type feedback was the least popular while comparison with historic data was the most popular. The analysis of the response is shown in Figure 5-3.

The results are close which raise a question about the way the question was structured. This was supported by the qualitative analysis of the user interface question in which the respondents were asked to comment on a suggested screen saver as means of electricity feedback.

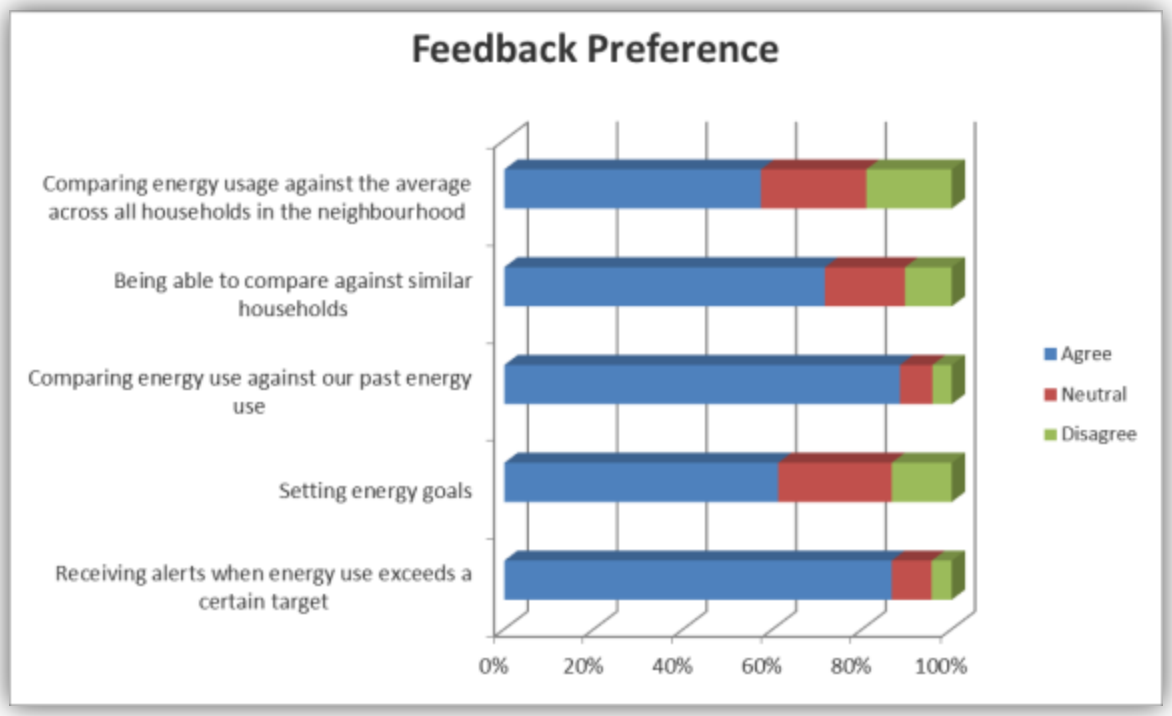


Figure 5-3 Types of feedback preference survey response

The themes arising from analysis in NVIVO are shown in Figure 5-4 and included suggestion of graphical representation of data, historical data and feedback in monetary and energy forms. The question and the pie chart representation of the quantitative answers are shown in Figure 5-4 and Figure 5-5 respectively.

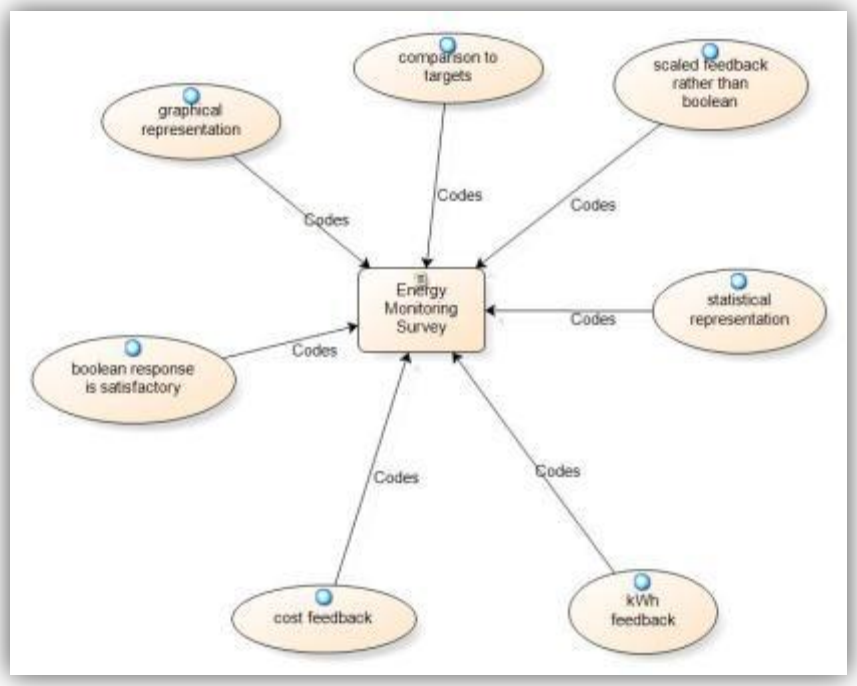


Figure 5-4 Themes recurring in feedback preference

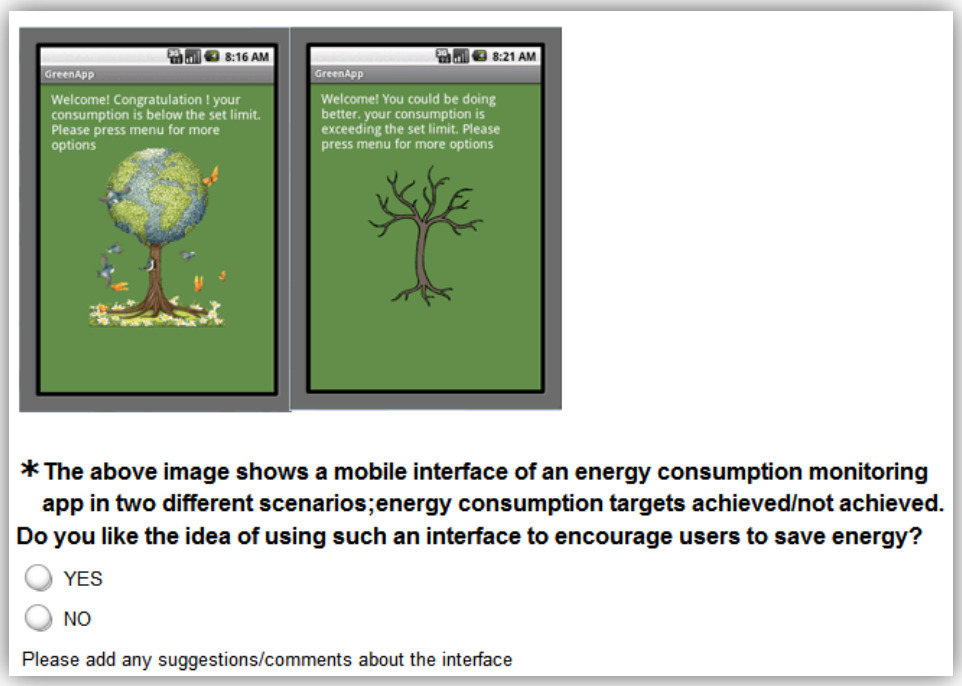


Figure 5-5 mobile interface and feedback methods survey question

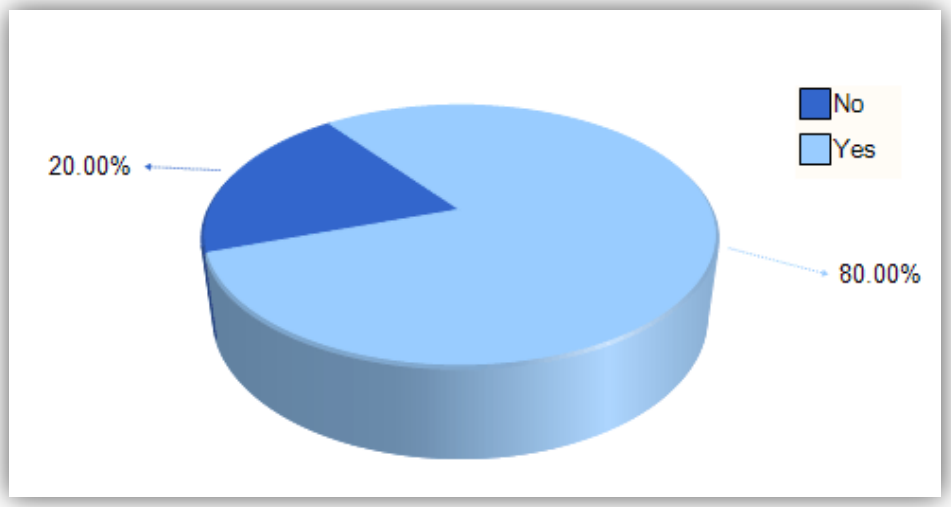


Figure 5-6 Mobile interface and feedback methods survey response

Although 80% of the participants said they liked the idea of the interface (refer to Figure 5-6). The qualitative analysis using NVIVO of the suggestion/comment section showed interesting themes

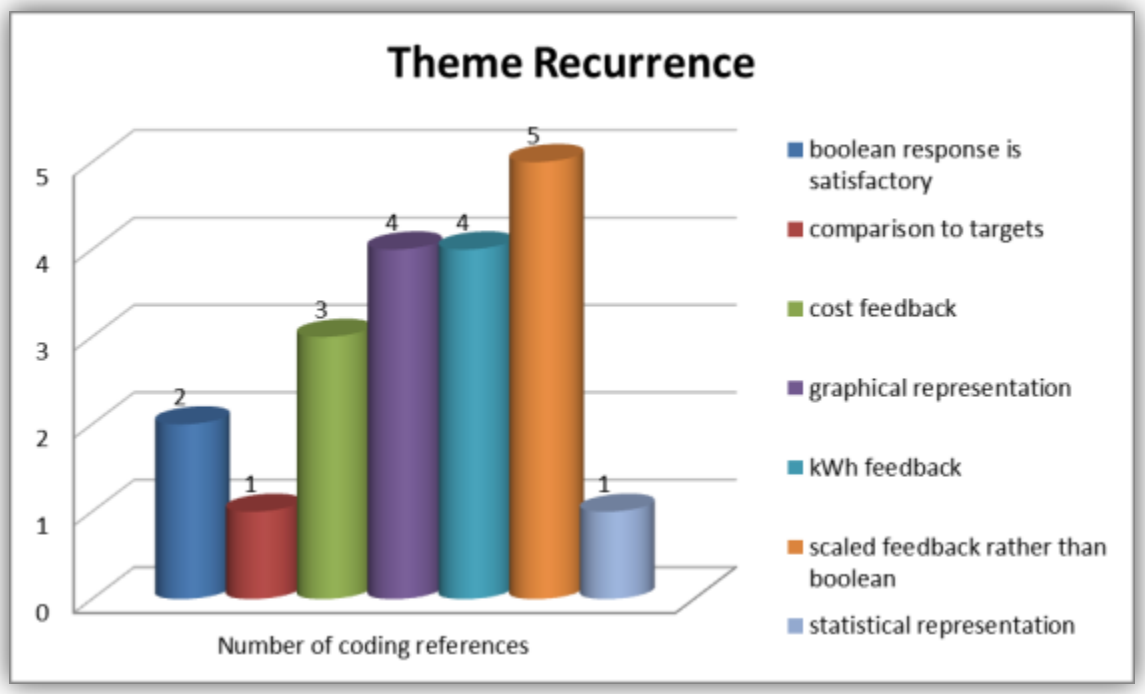


Figure 5-7 Theme recurrence in the mobile interface and feedback survey question

Views on Environmental Issues

When asked to rate the important of 65% of whom thought that environmental issues are quite important whilst 26% thought that they are very important. On the contrary, 4% said that environmental issues are not important and an equal percentage had no opinion on the matter.

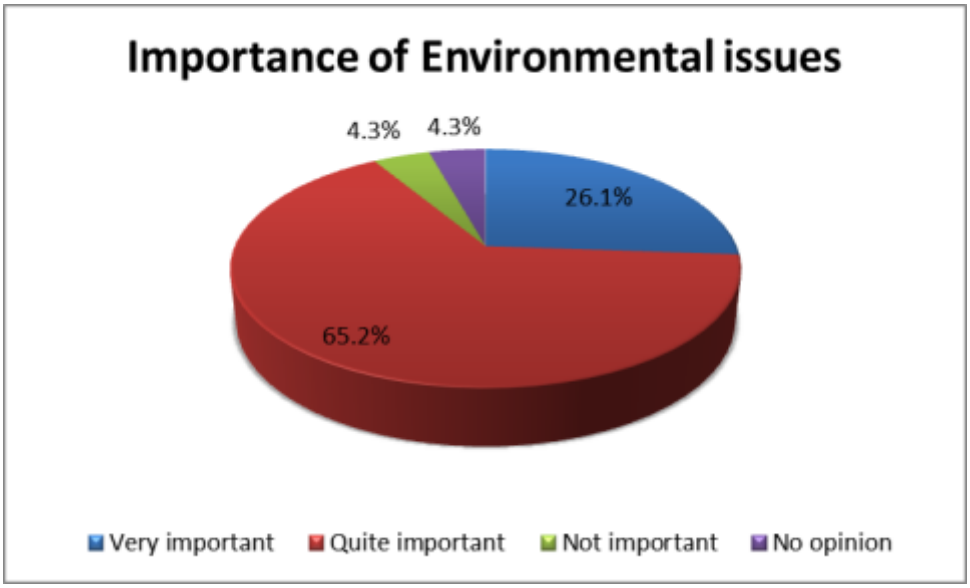


Figure 5-8 Views on environmental issues survey response

The cylinder chart in Figure 5-9 represents the frequency of recurrence of each of the themes considered when the above question was analysed.

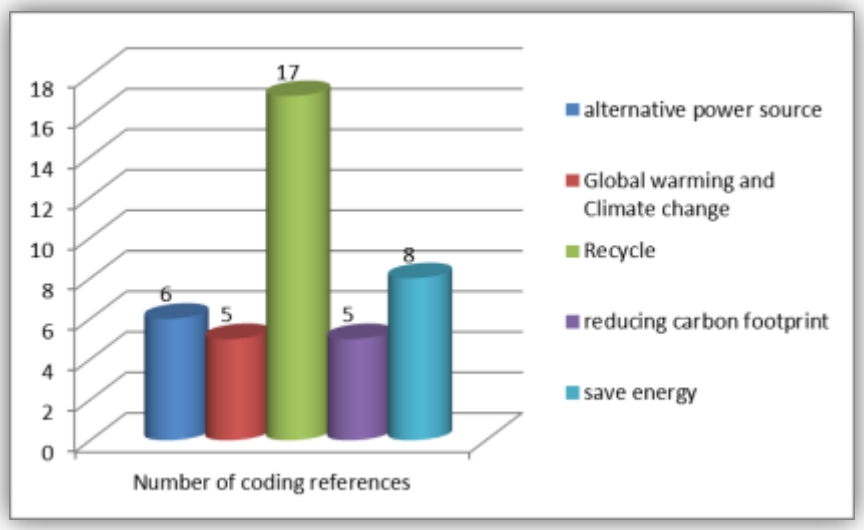


Figure 5-9 Theme recurrence in environmental views survey question

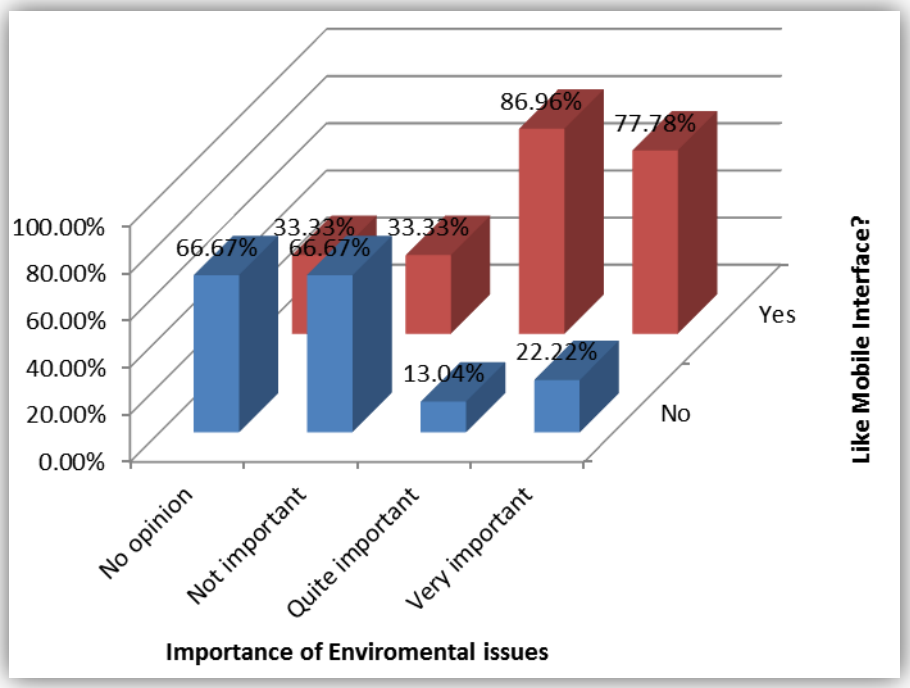


Figure 5-10

Figure 5-10 illustrates the correlation between environmental enthusiasm and the response to the mobile interface suggestion. The fact that more environmentally concerned participants liked the user interface demonstrates that the interface might have been unintentionally designed to target that category.

5.2 Interviews, Observations and Trials

5.2.1 Overview

The three research methods were used in conjunction with each other. Firstly, each of the participants was interviewed; the pre-interview drew a picture of the energy consumption behaviour of the individual's household as well as their preferred method of energy feedback. Secondly, the electricity meter readings of each participant's household were recorded twice with a week gap to measure the electricity usage before the system was installed; this was used as a baseline for energy consumption analysis and comparisons.

Thirdly, the system was installed in participants' houses and the mobile application was installed on their handheld devices through the Android market. Real time data measured by the energy monitoring system was observed through the researcher's access to the same remote platform feed.

Finally, a second interview with each of the participants was conducted one week after deploying the system. This aimed to investigate the effect of using the system on the participant's energy consumption habits, cover feedback from participants on the GreenAPP system as well as any suggestions for improvements.

The interviews were piloted with a colleague to ensure that the right questions are asked and enough time is allowed to obtain relevant, meaningful data. The questions of the interview were also aimed at testing some of the interesting results of the online survey, mainly; collecting detailed information about the incentives that encourage individuals to save energy.

5.2.2 Analysis and Findings

Household's Electricity Consumption

The electricity meter readings taken throughout the trial are shown in Table 5-1. These reading were used to create the energy versus time graphs for each household shown in Figure 5-11 and Figure 5-12. The graphs were obtained by subtracting each two consecutive readings and dividing that by the difference in days between the two corresponding dates, this generated the average daily consumption for each household throughout the trial.

Household A					
Date	10/08/2011	18/08/2011	06/09/2011	12/09/2011	22/09/2011
Reading	13752	13877	14078	14165	14298
Household B					
Date	12/08/2011	19/08/2011	22/08/2011	29/08/2011	
Reading	04066	04160	04197	04265	

Table 5-1 participating households' meters readings

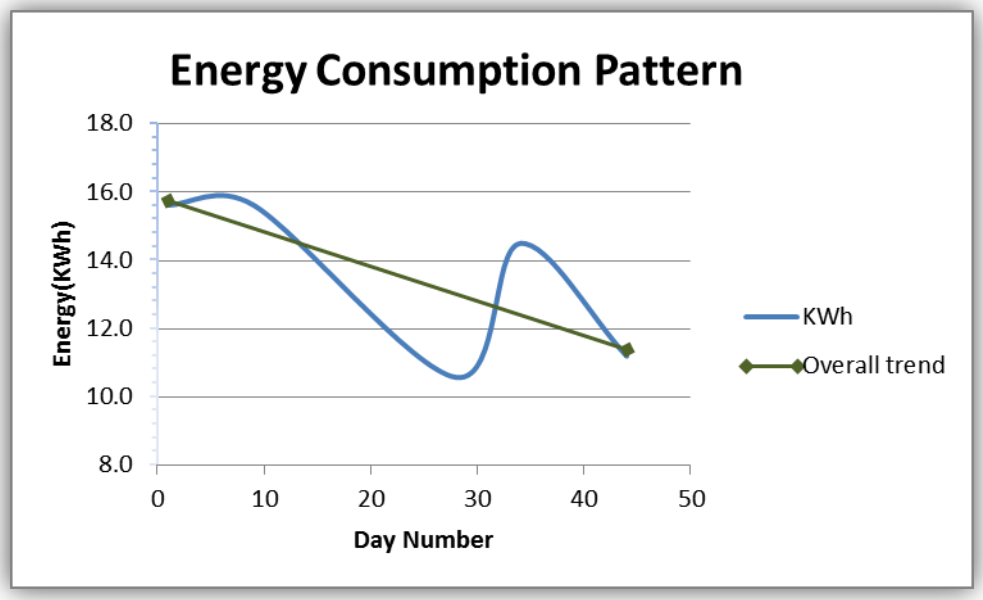


Figure 5-11 house A energy vs. time

Note: Household A predicted they would not have a decrease in their energy consumption as they had an extra person coming to live with them half way through the trial. They have also been on holiday after the first set of readings was taken.

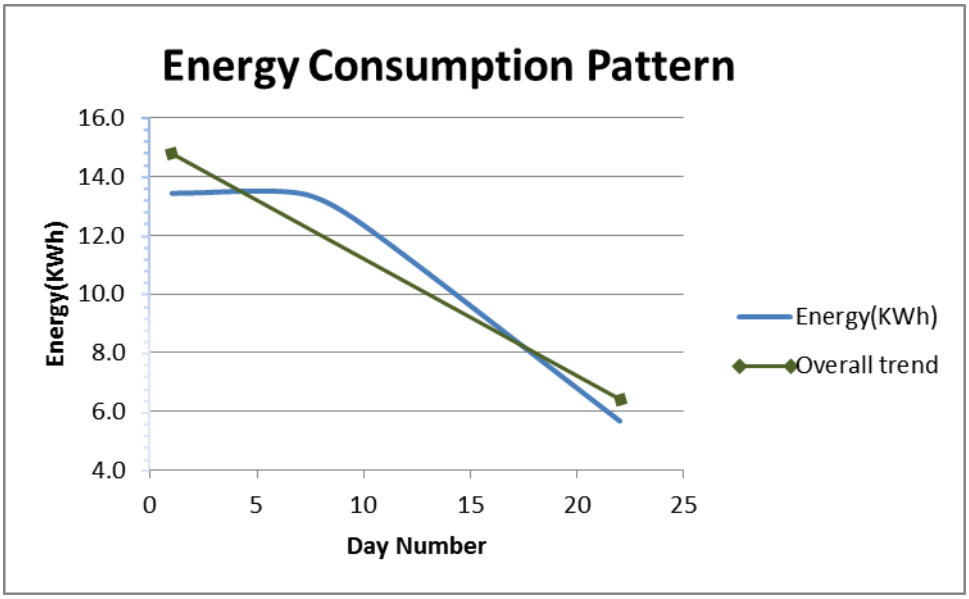


Figure 5-12 House B energy vs. time

In general, an overall drop in energy use has been noticed, however the limited time allocated to conducting the trial limits generalisation.

Interviews

Two interviews were held with each of the participating household representatives. The first interview aimed to explore participants views and practice regarding energy saving and ecological issues. This interview formed a baseline with which the second interview is compared and potential attitude/behavioural change detected. The post-trial interview had the same purpose as the former but with the added advantage of getting users feedback after they have used the energy monitoring system and the mobile application. The usability feedback part of the second interview is presented in section 6.1. The pre-trial and post-trial interview questions can be found in Appendix A and Appendix B respectively. The consent form signed by interview/trial participants can be referred to in Appendix C

Interviews' Themes and highlights

NVIVO coding facilities were used to aid the analysis of both sets of interviews; the following themes were identified in the pre-trial interview:

- participants felt the trial will help them to change the way they use electricity
- Participants had little awareness of electricity consumed by appliances and what energy rating means.

More themes are shown in Figure 5-13

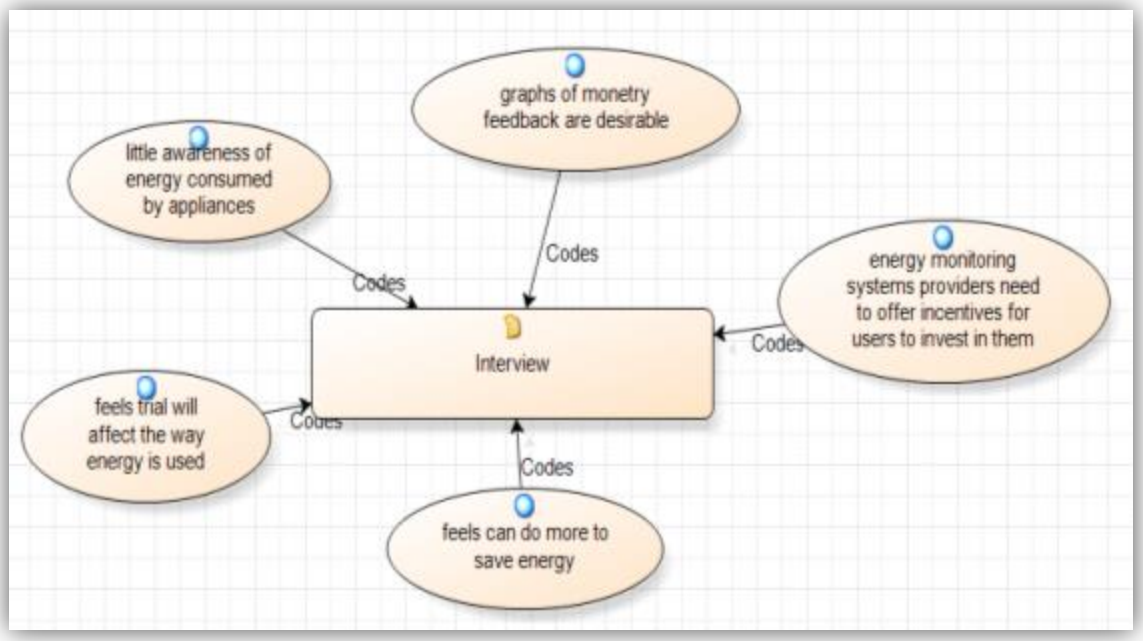


Figure 5-13 Pre-trial interviews main themes

Interesting post-trial Interview Findings

Participating in the trial has increased participants' awareness of the energy they use as well as the amount of money it cost one of the participants describes what happened when they started using GreenAPP to view historical usage.

"*It has raised my awareness significantly, and the ability to find information on my phone was very useful. I was over the moon to see billable use last month was only £22, and we have reduced our Direct Debit, and claimed s substantial amount of money back off our provider*"(Interviewee A)

And when asked about their understand of the information in GreenAPP they had the following response

"*Ok. It's been a while since I used KWh – not something I have paid much attention to since high school physics. I understand the concept but it did not feature heavily in my day to day thinking. Except now it does – a lot more. The graphs helped with the information a lot.*"(Interviewee A)

GreenAPP was modified constantly based on user feedback. One of the respondents had the following suggestion

"*I think I would have done better if I'd had chance to customise the name of the wall socket plus – to enable me to know that computer was "plug 3", and kettle was "plug 2" rather than having to remember plug 2 was "3" and plug 4 was "4". I sometimes got a bit confused*" (Interviewee B)

The application was modified to enable users to customise appliance names.

The general themes in the post-trial interview are shown in Figure 5-14

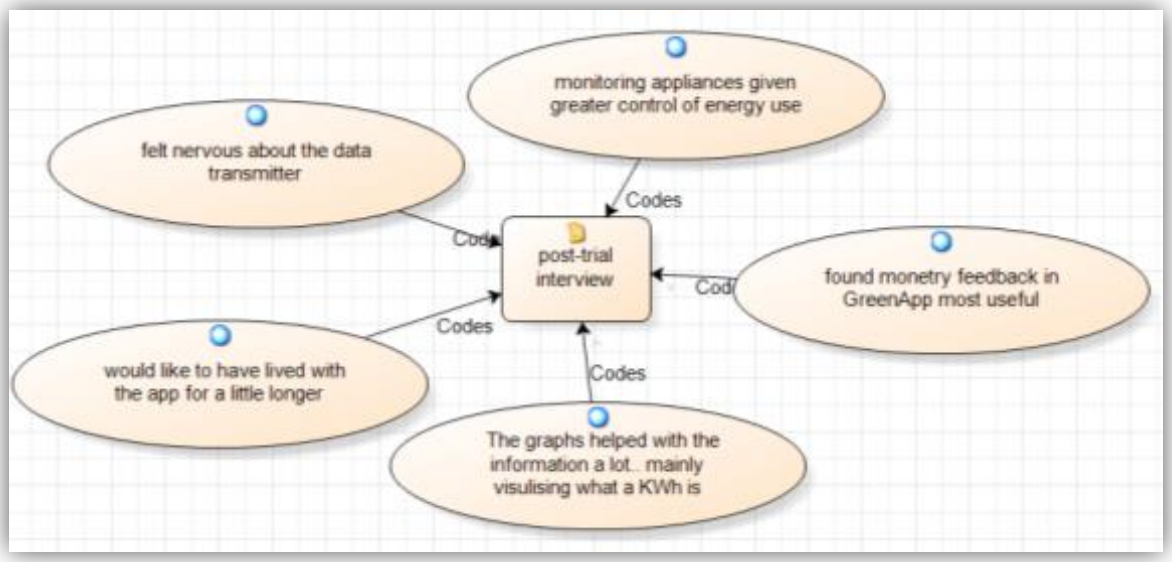


Figure 5-14 Post-trial interview main themes

5.3 Summary and Conclusion

This chapter presented the findings of the research carried out throughout the life span of this project. Online survey provided useful information about users preferring comparison with historical data as mean of feedback to make energy visible. The interviews responses backed up the finding of the survey analysis and provided helpful feedback to improving the mobile application.

Chapter 6 Testing and Evaluation

This chapter is divided into two main parts; the first part is testing which describes the methods and tools used to verify that the GreenAPP data transmitter and the GreenAPP mobile application operate in the desired manner and meet the requirements they were intended for. The second part of this chapter evaluates the overall approach to this project as well as each stage of it.

6.1 Testing

6.1.1 Purpose of Testing

In order to provide an acceptable user experience, GreenAPP was tested throughout the development stage. Testing was used to verify and validate GreenAPP and the data transmitter. Different scenarios or test cases were used to test performance, robustness and behaviour of each function within each of the applications

6.1.2 Testing Techniques

The testing techniques used in this project varied depending on the application being tested. The classic C based program used to drive the embedded device transmitter and the classic Java based program used to transmit data from a computer were tested using debugging facilities in the IDEs they were developed in. the Android has its own testing framework which was used to test GreenAPP. It was also tested using debugging facilities to verify calculations. Usability testing was carried out by the trial participants and the wider public through the Android market.

Data Transmitter Testing

The embedded device circuit board have a USB connector which enables programs to be installed on the board as well as debugging messages to be sent from the board to the connected computer. In order to verify the average power calculations described in section 4.1.2, the serial port is used to output the data received from the monitor's central control unit and the data transmitted to the remote server. This is done by modifying the program to open a connection to the serial port and printing the required data to it as shown in the code snippet below.

```
Setup(){
     ...
     Serial.begin(57600);
}
readData{
     ...
     Serial.print("data received:");
     Serial.println(data);
}
writeData{
     ....
     Serial.println("data transmitted: "+dataT);
}
```

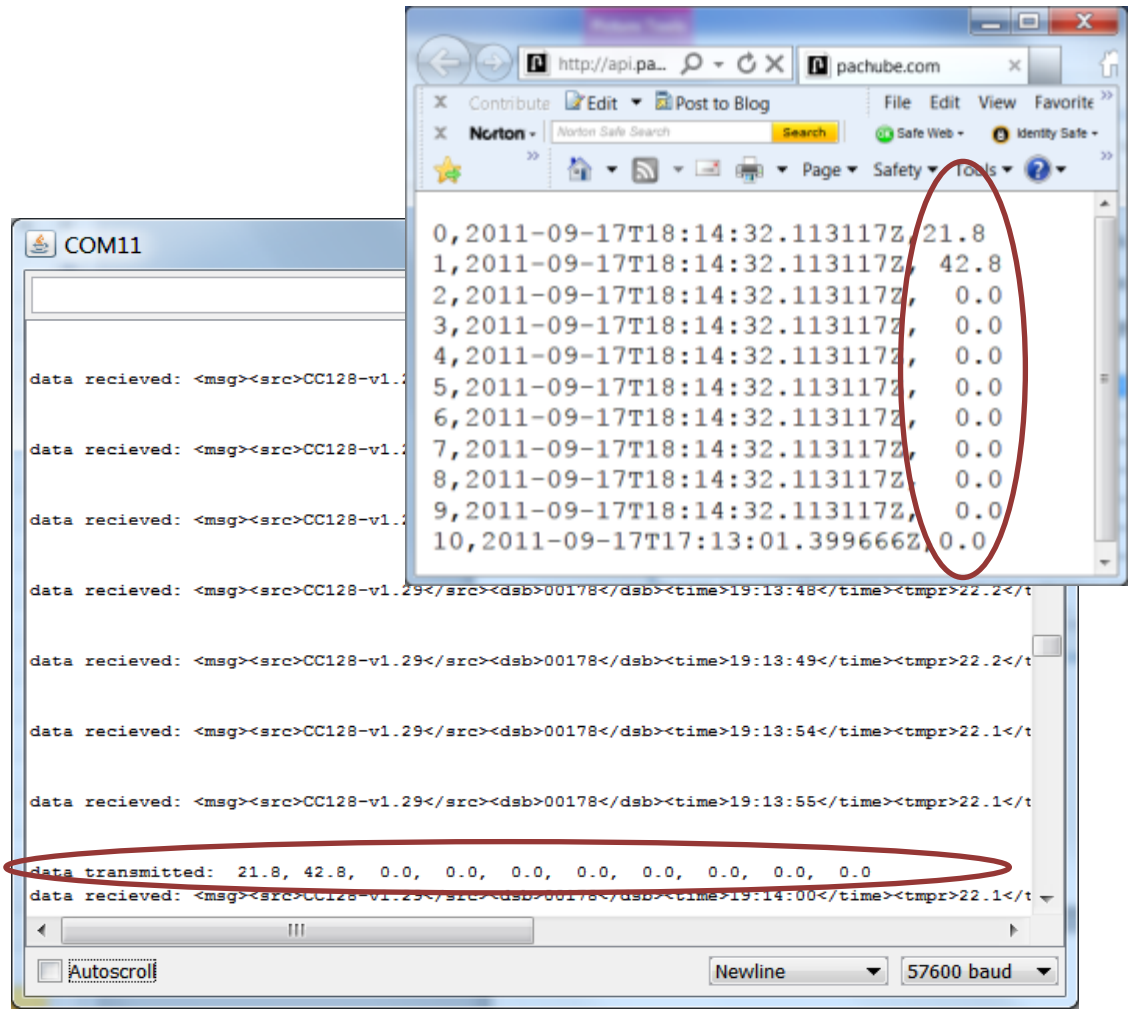


Figure 6-1 Serial port display connected to the embedded device vs. remote server current data

This data is copied and organised to extract the power and temperature readings. The extracted data are then processed in Excel to calculate the averages. The calculations are then compared with data sent to the remote web server as well as using the remote server API to get the data received by it; a comparison between the two is shown in

Figure 6-1 . This can also be verified by the data displayed on GreenAPP's main screen seconds later as Figure 6-2 illustrates.

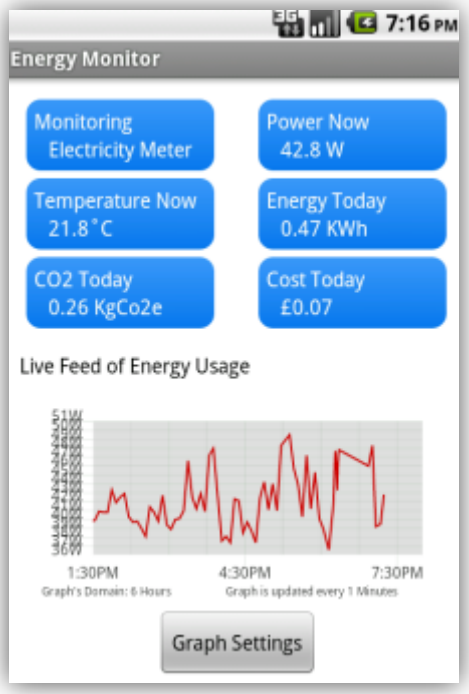


Figure 6-2 GreenAPP Display verification of the data transmitter functionality

Testing the GreenAPP Mobile Application

Android Platform Testing facilities

Android SDK has a variety of useful tools to aid the testing process of the developed application. The tools used while developing GreenAPP are

- The mobile device emulator: this is a virtual mobile device and has the same functionalities of an actual mobile phone. The emulator specifications can be changed such as the SDK RAM, the screen resolution, the API which made it an invaluable tool to testing GreenAPP and its suitability for running on a variety of different handsets. (See Figure 6-3)
- Dalvik Debug Monitor Server DDMS: have been used in conjunction with the emulator to monitor current operations and access debugging data. (See)
- A Test project was devised to test the different elements of the application using JUnit.

Figure 6-3 Android Emulator

System Testing

The application testing occurred throughout the development process. The debug mode of eclipse was instrumental in this process. Using facilities like breakpoints and "variable watch" provided the ability to trace the single data element journey through the system.

Using DDMS and the Android Log class, data values were tracked and verifying calculations performed Excel spread sheets.

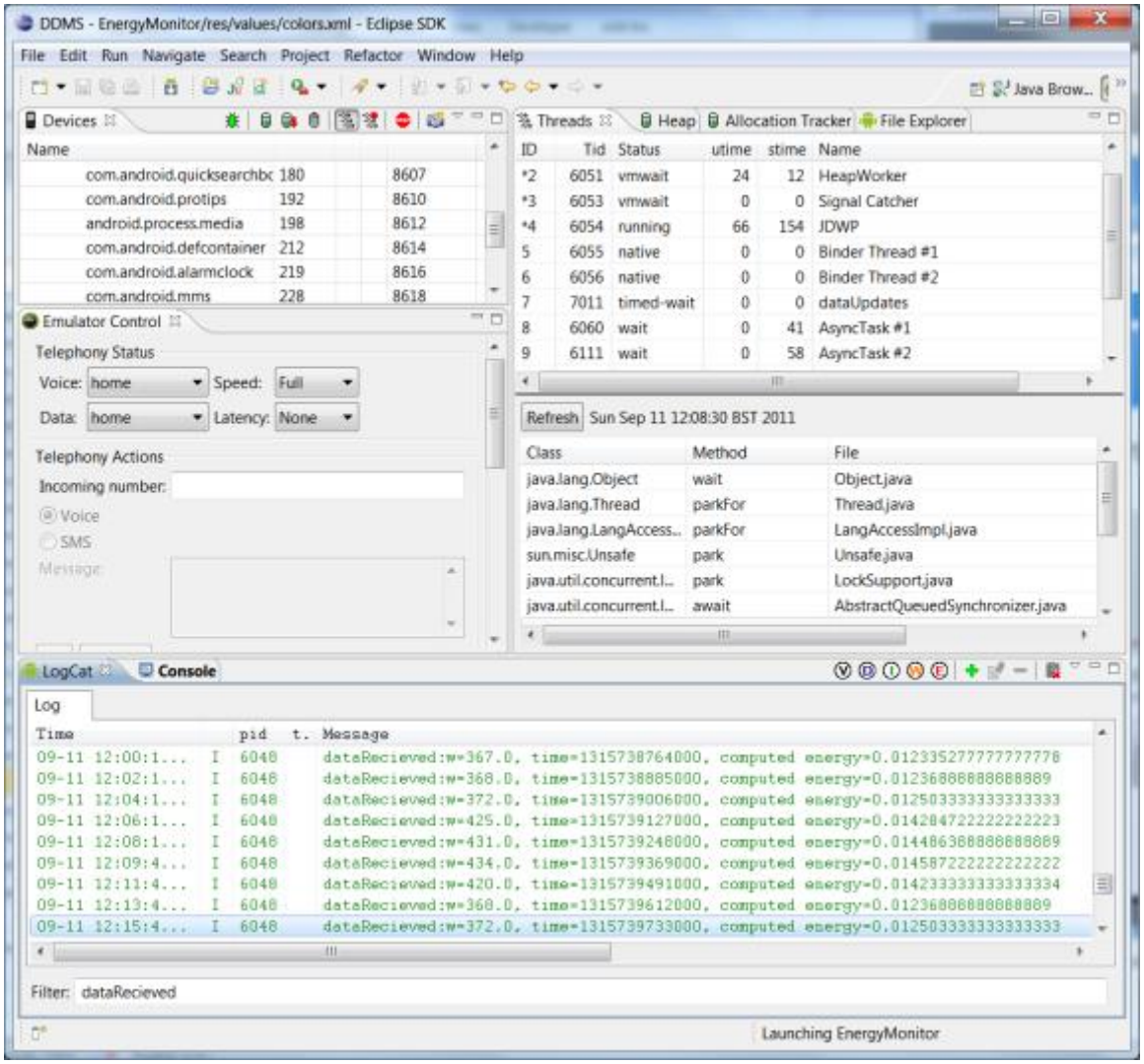
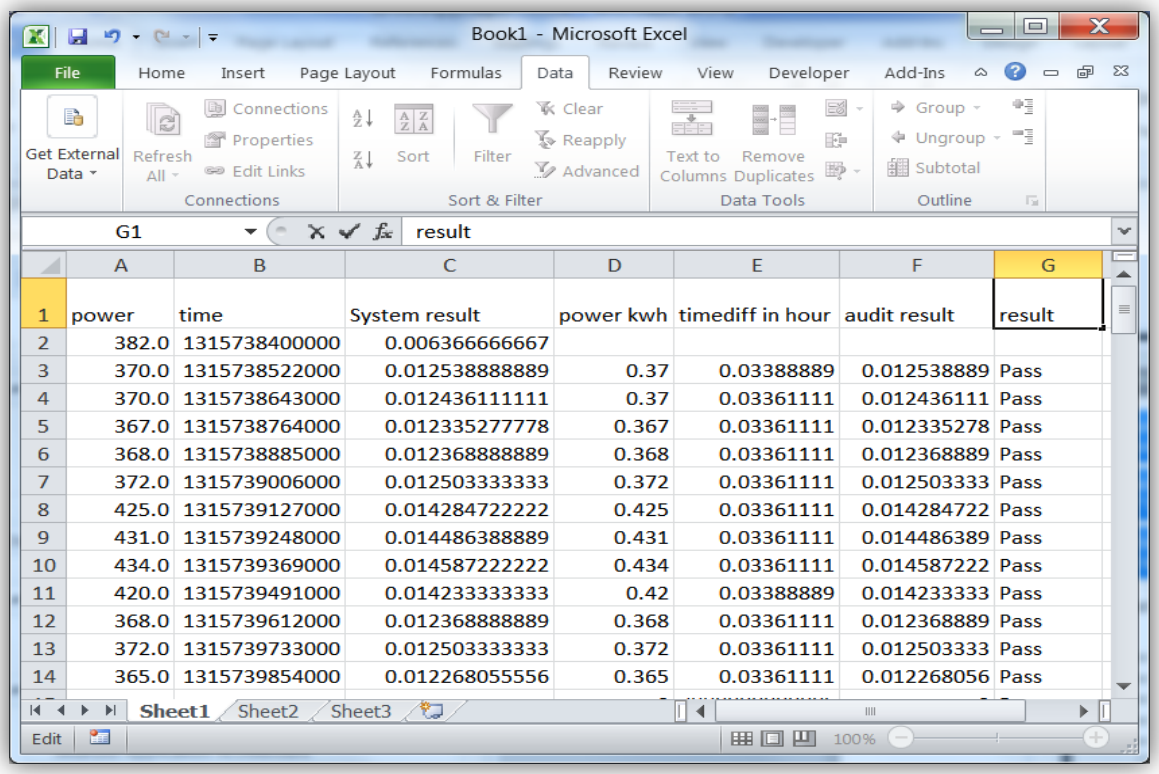


Figure 6-4 Dalvik Debug Monitor Server DDMS



	A	B	C	D	E	F	G
1	power	time	System result	power kwh	timediff in hour	audit result	result
2	382.0	1315738400000	0.006366666667				
3	370.0	1315738522000	0.012538888889	0.37	0.03388889	0.012538889	Pass
4	370.0	1315738643000	0.012436111111	0.37	0.03361111	0.012436111	Pass
5	367.0	1315738764000	0.012335277778	0.367	0.03361111	0.012335278	Pass
6	368.0	1315738885000	0.012368888889	0.368	0.03361111	0.012368889	Pass
7	372.0	1315739006000	0.012503333333	0.372	0.03361111	0.012503333	Pass
8	425.0	1315739127000	0.014284722222	0.425	0.03361111	0.014284722	Pass
9	431.0	1315739248000	0.014486388889	0.431	0.03361111	0.014486389	Pass
10	434.0	1315739369000	0.014587222222	0.434	0.03361111	0.014587222	Pass
11	420.0	1315739491000	0.014233333333	0.42	0.03388889	0.014233333	Pass
12	368.0	1315739612000	0.012368888889	0.368	0.03361111	0.012368889	Pass
13	372.0	1315739733000	0.012503333333	0.372	0.03361111	0.012503333	Pass
14	365.0	1315739854000	0.012268055556	0.365	0.03361111	0.012268056	Pass

Figure 6-5 example of data verification using Excel

The computer-based data transmitter GUI was also used to verify the functionality of the `DataService`

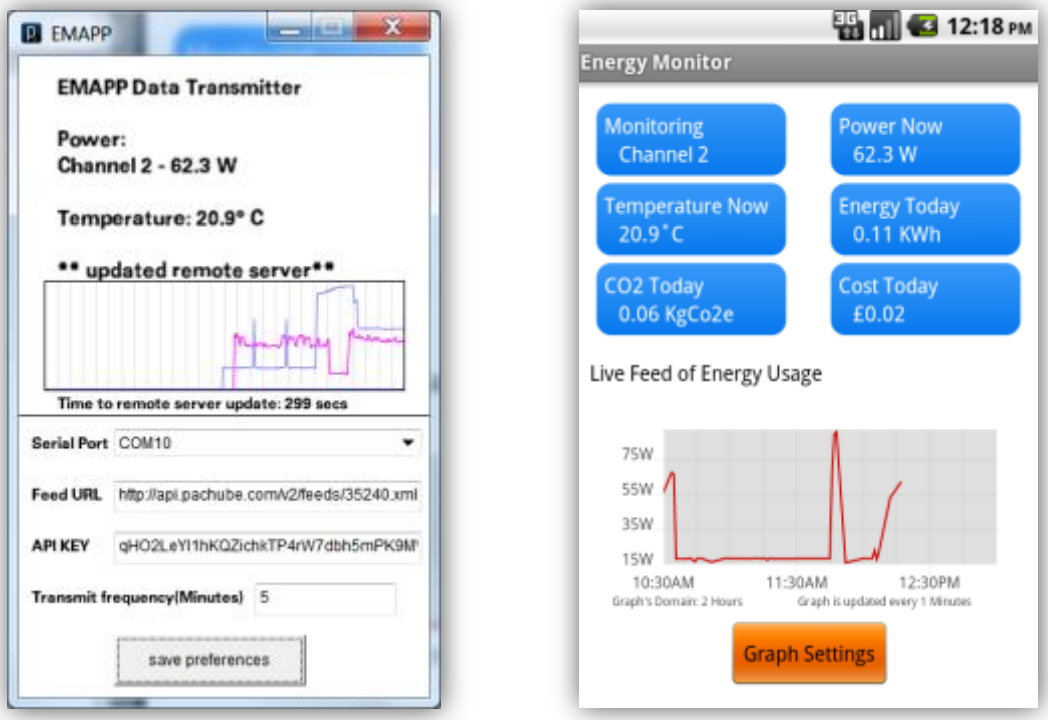


Figure 6-6 Data Sent/Data Received Comparison

User Testing

The GreenAPP was also published on the Android market and made available to download all over the world for any interested enthusiast free of charge. This provided a means of fixing bugs and behavioural problem in GreenAPP and automatically updating the version of it running on the trials participants' smartphones. This is achievable through the error reporting mechanism in Android. The testing of GreenAPP has been a complicated task because of the live feed and graph updates facilities. Through the reporting mechanism some problems that depended on the time of the day or the internet connection availability were detected and rectified. Figure 6-7 shows the report panel for GreenAPP.

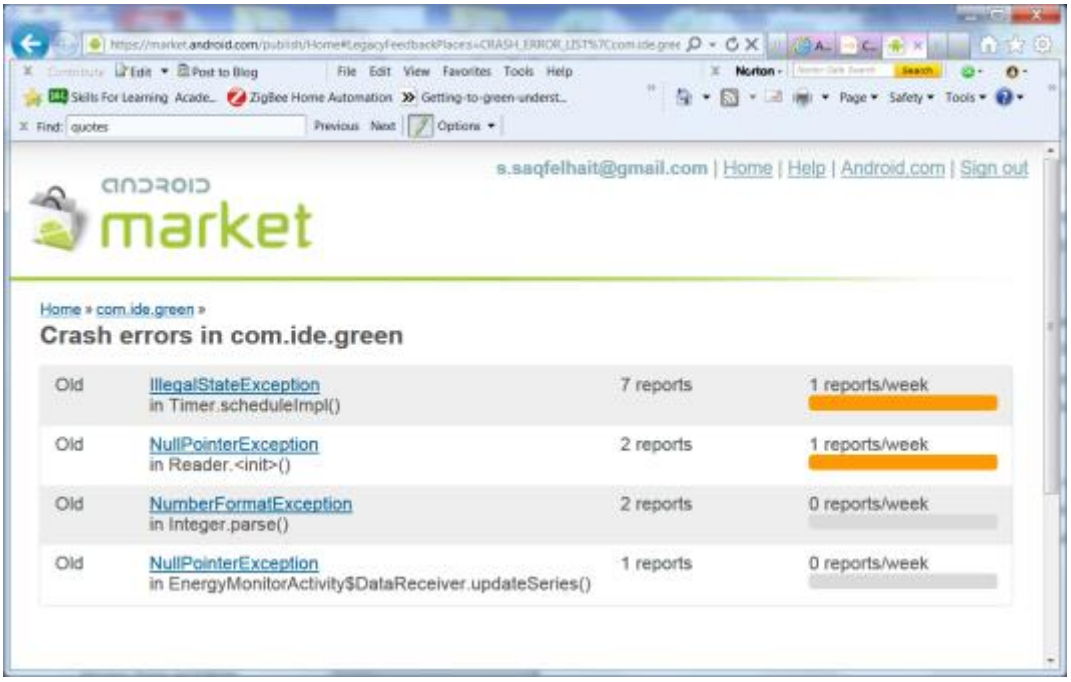


Figure 6-7 Android Market reports for GreenAPP

Figure 6-8 shows the statistics from Android market of platforms on which GreenAPP has been deployed.

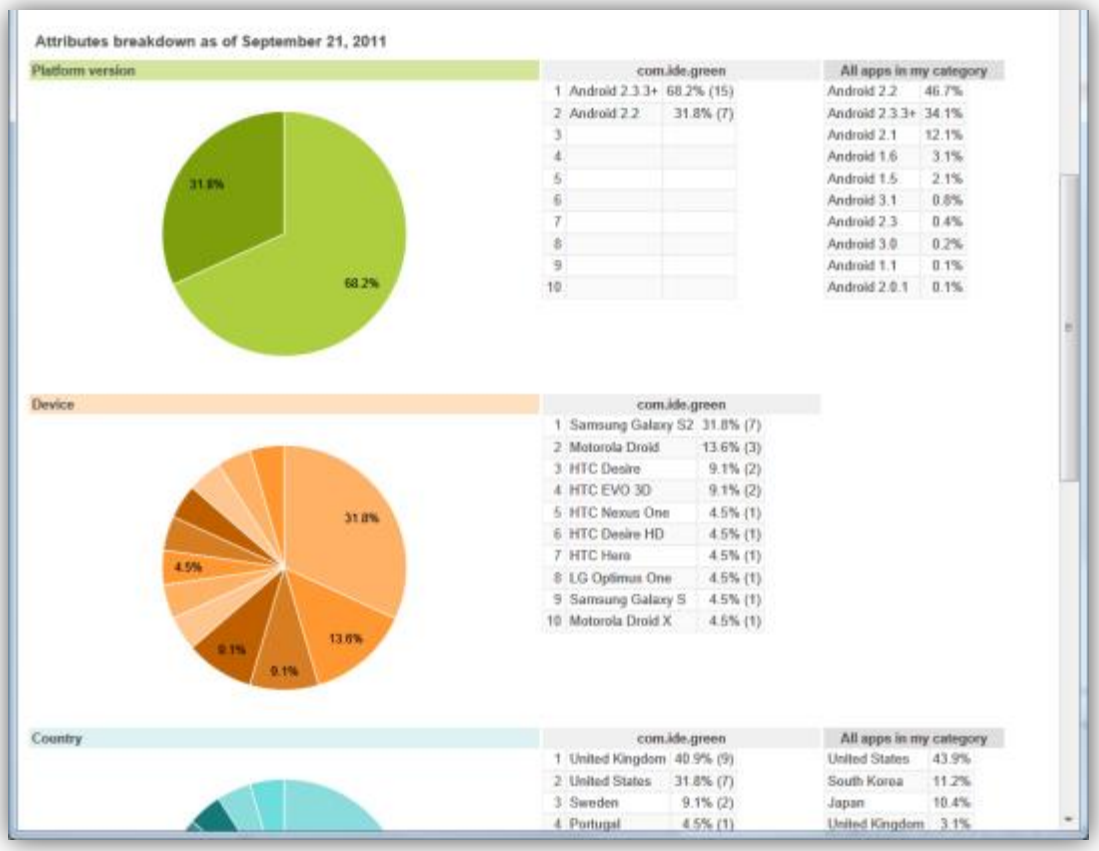


Figure 6-8 Deployment platforms for GreenAPP

Trial participants tested the GreenAPP which resulted in many changes to the interface. The following are some examples

- Main screen user interface text fields looked like highlighted buttons
- x-y axes range and step values changed as described in section 4.2.4
- list of appliances did not work on touch screen only handsets which resulted the change of the way that activity was implemented

In the post-trial interview participants were asked about main aspect of the application usability such as clarity of information, ease of use, intuitiveness of the navigation system and the actual interface. GreenAPP was interface rated 4/5 by the participants. The interface has been described "I did not notice it, which means it's good!"(Interviewee A).

The following figures show some of the validation carried out while testing GreenAPP. Figure 6-9 demonstrate the editing appliance process in action and Figure 6-10 illustrates the input validation process of the Price Activity.

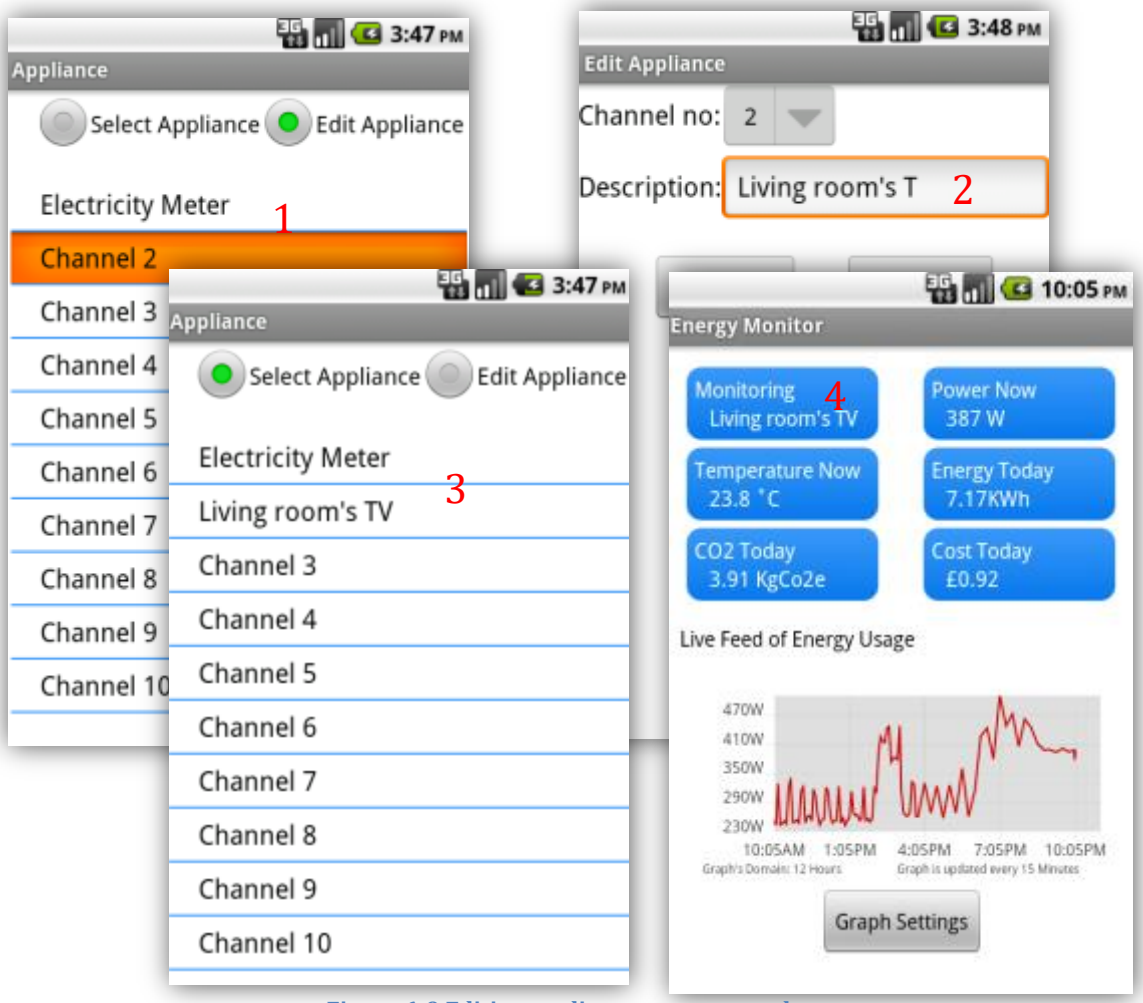


Figure 6-9 Editing appliance name example

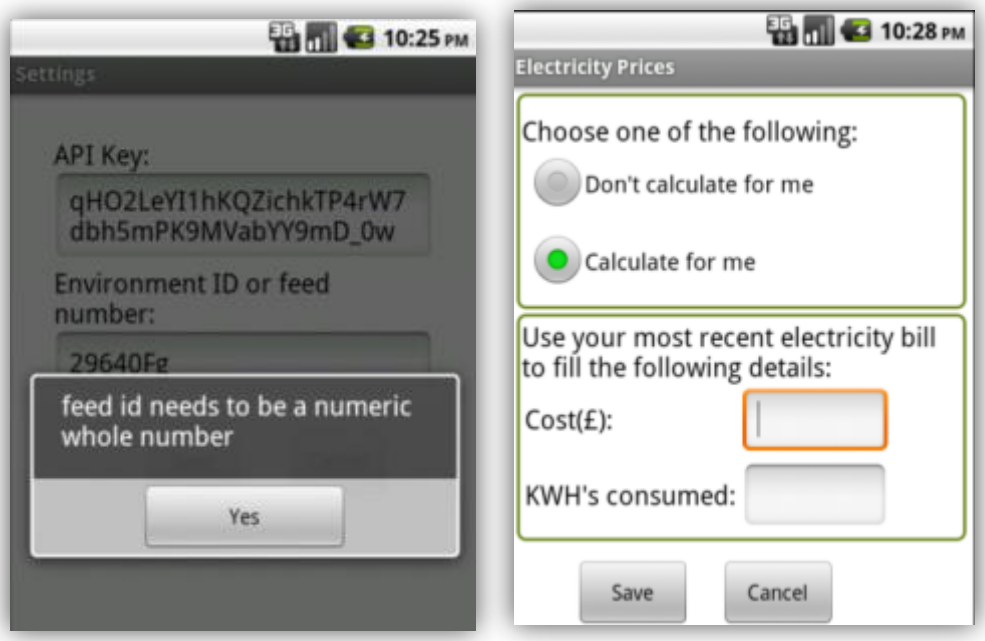


Figure 6-10 Checking for invalid input

6.2 Methodology and Project Evaluation

The strategy used in the evaluation of GreenAPP is the participant-oriented model; this involved the potential users in the every stage of the development cycle. The previous section demonstrated the testing strategy used to test GreenAPP. In this section the outcome evaluation is used to evaluate the research methodology whilst impact evaluation is used to evaluate the overall project.

Choosing the research methodology for this piece of work was based on thorough literature review in the related fields of research. The online survey offered very informative data about the application's target audience, possibilities of use and the ways in which participants preferred to be fed back to about their energy usage. It also affected the design decisions of GreenAPP through user reactions to the user interface encompassed in the survey. This has provided a means of evaluation the design before implementing it, saving potential problems in the implementation stage.

Although the project is complete, the time available scale envisaged in the project plan was stretched; this is due to unforeseen complications with the energy monitoring system. In the requirement gathering stage, all the devices needed to conduct the trial and aid the development of GreenAPP were purchased except the IAMs which were out of stock. This delayed the testing of the monitoring system and when the IAM became available an unexpected problem emerged; the transmitting device procured did not support the transmission of appliance data. This led to the development of the computer-based data transmitter which, on reflection, was unnecessary. At that point, the problem with the computer-based data transmitter became obvious; it defeats the purpose of the whole project, potential trial participants felt uneasy about keeping a computer 24/7 on in order to monitor their energy use which they are trying to reduce. This is when the idea of the embedded device data transmitter developed.

The time limitation limited also the options about the remote server and minimising the resources GreenAPP uses on the users handset. The ideal scenario would be to have a dedicated server with a RESTful server side script such as Java Servlets to handle historical data enquiries and minimise calculations performed by the application. In addition, the use of a third party server for data storage/retrieval could cause security issues.

The data collected from the interviews, complemented the survey outcomes, however, the two interviews placed a lot of constrain on the researcher's and participants' time. A post-trial survey might have been sufficient.

6.3 Summary and Conclusion

This chapter established the techniques used to evaluate GreenAPP and the overall research. A variety of test cases of the application were illustrated and users evaluation of the application was outlined

Chapter 7 Conclusion and Further Work

In this chapter, the contributions and conclusions of the work in this dissertation, as related to the research questions posed in section 1.1. This chapter ends with an indication of future research directions.

7.1 Conclusion

This research aims to investigate the use of ubiquitous technology such as mobile devices in a persuasive domain. Three main fields of research were explored; Captology, UbiComp and Domestic Energy Consumption. The scope of the research was very broad and the practical element of it was huge with a very limited time. The major conclusions are summarised as follows:

- Participants in the online survey showed a lot of interest in the project and contributed useful ideas in the comments section of most questions.
- The analysis and findings of the questions show a general public awareness of the environmental issues and a particular interest in recycling which can be related to the mass media coverage in the last few years. Respondents also have shown an interest in saving energy not only for environmental reasons but also for financial reasons. The most popular means of reducing energy invisibility was the use of comparisons with historical use data which was used in the implementation of GreenAPP.
- GreenAPP was developed, tested and evaluated using a user-centred approach. It was also launched to the Android market.
- Trials and interviews: although the number of participants is not large, their contributions in the first interview were supportive of the results of the survey. They were cautious of the energy monitoring system and GreenAPP prior to deployment because they did not know what to expect. However, once the system was installed they were very enthusiastic and generally impressed.
- The analysis of the trial's power consumption figures shows a general decrease in the average daily consumption of the participating households. In addition, the partakers indicated that having the energy monitoring system without the GreenAPP was not as helpful in saving energy as having both. This answers the research question in section 1.1 to a certain extent, since these results cannot be

generalised given the number of households taking place and the duration of the trial.

- An embedded device data transmitter has been assembled and programmed opening a business opportunity for the project.

Single appliance energy monitoring, the particular interest of this study has been explored in the trial but there is still a lot more to be done in this arena. The next section outlines general research directions

7.2 Further Work

Further work or research directions can be summarised in the following:

- Data Transmission and collection
    - Energy Monitoring Systems provider control the format of the data sent to the cloud, an investigation into a standard for energy monitoring data transmission could be helpful to provide the opportunity for mobile developers to create a variety of applications to cater for different users' taste and preference
    - The amount of data sent by energy monitoring systems is massive as discussed in section 4.1.3. A study on the best way of optimising, storing and sampling the data would be a great step in attempting to create a persuasive ubiquitous energy monitoring device.
    - Improve the PBC based data transmitter through better connectors

- Single Appliance Energy Consumption
    - Control of the energy consumption of the appliance by introducing the ability to switch the appliance on or off using GreenAPP either by the user or by the application according to set targets.
    - A comprehensive trial that looks at tracking appliance energy consumption and participants' behaviour as consumption becomes more visible and concrete.

- Energy Consumption Research
  - Multidisciplinary research efforts through wider involvement from experts in psychology, graphic design, software engineering, etc. would contribute in a better way to persuasive technology with regards to the best ways of feedback would benefit this field.
  - Look at ways to make energy consumption units an easy term to understand and scale
  - Target specific groups of the population who are not particularly enthusiastic about the subject
  - Explore wider field of use such as business, public sector and third sector and the practical deployment of similar systems in these environments.

- Mobile Applications Development
  - Look at clear methodology for developing mobile application from software engineering point of view,

References

Android (n.d.) Application Fundamentals | Android Developers [Internet]. Available from: <http://developer.android.com/guide/topics/fundamentals.html> [Accessed 13 September 2011].

Arduino (n.d.) Arduino - Reference [Internet]. Available from: <http://arduino.cc/en/Reference/HomePage> [Accessed 6 September 2011].

Björkskog, C.A. et al. (2010) EnergyLife: Pervasive Energy Awareness for Households. In: **Ubicomp '10 Proceedings of the 12th ACM international conference adjunct papers on Ubiquitous computing**. Copenhagen, ACM, pp.361-362.

Bos, B. et al (2007) Remote Management and Secure Application Development for Pervasive Home Systems Using JASON. In**: Third International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing SecPerU**. pp. 7-12.

Bourner, T. (2002) The Research Journey: Four Steps to Success. In: Greenfield, T. **Research Methods for Postgraduates**, 2nd Ed. London, Arnold, pp.7-12.

Bryman, A., 2006. **Integrating quantitative and qualitative research: how is it done? Qualitative Research**, 6(1), pp.97-113.

Butler, M. (2011) Android: Changing the Mobile Landscape. **IEEE Pervasive Computing**, 10, p.pp.4-7.

Canalys (2011) Google's Android becomes the world's leading smart phone platform (Canalys research release: r2011013) [Internet]. Available from: <http://www.canalys.com/pr/2011/r2011013.html> [Accessed 14 June 2011].

Carbon Trust (2011) Conversion factors - Energy and carbon conversions. Available from: <http://www.carbontrust.co.uk/publications/pages/publicationdetail.aspx?id=CTL113> [Accessed 9 August 2011].

Carroll, J.M. (2001) Introduction: Human-Computer Interaction, the Past and the Present. In: Carroll, J.M. **Human Computer Interaction in the New Millennium**. New York. ACM Press pp. xxvi-xxxvii.

Chao, K.M et al (2010) A Profile based Energy Management System for Domestic Electrical Appliances. In: **IEEE International Conference on E-Business Engineering**. November, California, IEEE Computer Society. pp. 415-420.

Charland, A. & Leroux, B., 2011. Mobile application development: web vs. native. **Communications of the ACM**, 54(5), pp.49-53.

Chetty, M., Tran, D. & Grinter, R.E. (2008) Getting to green: understanding resource consumption in the home. In**: Proceedings of the 10th international conference on Ubiquitous computing**. Seoul, Korea, ACM, pp.242-251.

Cockburn, A. & Highsmith, J. (2001) Agile Software Development: The People Factor. *Computer*, 34, p.pp.131-133.

Conway E. (2008) What's in a Name? Global Warming vs. Climate Change [Internet]. Available from: <http://www.nasa.gov/topics/earth/features/climate_by_any_other_name.html> [Accessed 10 June 2011].

Denscombe, M. (2007) **The good research guide: for small-scale social research projects**, 3rd ed., Open University Press.

Department of Energy & Climate Change (2009) Digest of United Kingdom Energy Statistics Table 1.1.6: Expenditure on energy by final user to 2009 [Internet]. Available from: <http://www.decc.gov.uk/assets/decc/statistics/source/total/dukes1_1_6.xls> [Accessed 14 November 2010].

Dix, A. et al. (2004) **Human-Computer Interaction.** 3rd ed. England, Pearson Education.

Fallman, D. (2008) The Interaction Design Research Triangle of Design Practice, Design Studies, and Design Exploration. **Design Issues**, 24 (3), pp.4-18.

Fitzpatrick, G. & Smith, G. (2009) Technology-Enabled Feedback on Domestic Energy Consumption: Articulating a Set of Design Concerns. **IEEE Pervasive Computing**, 8, pp.37-44.

Fling, B. (2007) **Mobile Design and Development**. 1st Edition. O'Reilly.

Fogg, B.J. (1997) Captology: the study of computers as persuasive technologies. **CHI '97 extended abstracts on Human factors in computing systems: looking to the future**. Atlanta, Georgia, ACM: 129-129.

Fogg, B.J. (2002) Persuasive technology: using computers to change what we think and do. **Ubiquity**, 2002(December), pp.2.

Fogg, B.J. (2009) Creating persuasive technologies: An eight-step design process. In: **Proceedings of the 4th International Conference on Persuasive Technology**. Claremont, California, ACM

Froehlich, J. et al. (2009) UbiGreen: investigating a mobile tool for tracking and supporting green transportation habits. In: **Proceedings of the 27th international conference on Human factors in computing systems**. Boston, MA, USA, ACM, pp.1043-1052.

Gartner (2010) Gartner Says Worldwide Mobile Phone Sales Grew 35 Percent in Third Quarter 2010; Smartphone Sales Increased 96 Percent [Internet]. Available from: <http://www.gartner.com/it/page.jsp?id=1466313> [Accessed 14 June 2011].

Grondzik, W.T. et al (2010) **Mechanical and Electrical Equipment for Buildings**. 11th ed. New Jersey, Jon Wiley & Sons.

Hazas, M., Friday, A. & Scott, J., 2011. Look Back before Leaping Forward: Four Decades of Domestic Energy Inquiry. **IEEE Pervasive Computing**, 10, pp.13-19.

Hughes, M. (1996) Interviewing. In: **Greenfield, T. Research Methods: Guidance for Postgraduates**. London, Arnold, pp.169-177.

Lee, J.-S., Chuang, C.-C. & Shen, C.-C.(2009) Applications of Short-Range Wireless Technologies to Industrial Automation: A ZigBee Approach. In:**Advanced International Conference on Telecommunications**, 0, pp.15-20.

Pidgeon, C. & Ong, S. (1996) Chromatography: Gaussian-Shaped Chromatography Peaks. In**: Advanced Tutorials for the Biomedical Sciences: Animations, Simulations, and Calculation Using Mathematica**. New York, Wiley-VCH.

Pinkasovitch, A. (2010) The Changing Face Of Google (GOOG, MSFT, AAPL, YHOO, RIMM, AOL). Newstex Web Blogs [Internet blog]. Available from: <http://www.lexisnexis.com.ezproxy.leedsmet.ac.uk/uk/nexis/docview/getDocForCuiReq?perma=true&oc=00240&lni=5192-DY71-JCMN-Y1PC&csi=299488&secondRedirectIndicator=true> [Accessed 14 June 2011].

Saffer, D. (2009) **DESIGNING FOR INTERACTION Creating Innovative Applications and Devices**. 2nd ed. New Riders

Serhani, M. et al (2009), Toward an Efficient Framework for Designing, Developing, and Using Secure Mobile Applications**, Proceedings of World Academy of Science: Engineering & Technology**, 52, pp. 122-128

Shah, N., Tsai, C.-F. & Chao, K.-M. (2009) Monitoring Appliances Sensor Data in Home Environment: Issues and Challenges. In: **IEEE International Conference on Commerce and Enterprise Computing**, 0, p.pp.439-444.

Sharp, H., Rogers, Y. & Preece, J. (2007) **Interaction design: beyond human-computer interaction**. 2nd ed. Chichester, Wiley

Spagnolli, A. et al (2011) Eco-Feedback on the Go: Motivating Energy Awareness. In: **Computer**, vol. 44 no.5. IEEE Computer Society. pp. 38-45

Tompros, S. et al. (2008) A Pervasive Network Architecture Featuring Intelligent Energy Management of Households. In: **PETRA '08 Proceedings of the 1st international conference on PErvasive Technologies Related to Assistive Environment**. ACM.

Wasserman, A.I. (2010) Software engineering issues for mobile application development. In: **Proceedings of the FSE/SDP workshop on Future of software engineering research**. Santa Fe, New Mexico, USA, ACM, pp.397-400.

World Commission on Environment and Development,UN (1987) Our Common Future, Chapter 2: Towards Sustainable Development - A/42/427 Annex, Chapter 2 - UN Documents: Gathering a body of global agreements [Internet]. Available from: <http://www.un-documents.net/ocf-02.htm#I> [Accessed 14 November 2010].

Zapico, J.L. et al. (2009) Climate persuasive services: changing behavior towards low-carbon lifestyles. In: **Proceedings of the 4th International Conference on Persuasive Technology**. Claremont, California, ACM, pp.1-8.

Appendices

Appendix A **Pre- trial Interview Questions**

What are you currently doing to reduce your energy consumption?

Prompts

- Turn off all lights when you leave the room
- Unplug chargers from the mains
- Tumble dry your clothes
- Leave appliances on standby (e.g. TV, DVD player, games consoles)
- Boil just the amount of water you need when using the kettle? (e.g. making a cup of tea for one person)

Do you feel you are doing enough to reduce your carbon footprint?

What motivates you to save energy? (e.g. saving money, reducing carbon footprint)

Do you think that participating in this trial will have an impact on your energy consumption behaviour?

Do you have a clear understanding of your energy usage?

Suggestions:

- You carefully analyse the energy consumption of all your appliances and understand how each affects your total bill.
- You try to buy "green" appliances, but you don't really understand what it means.
- You carefully follow the environmental issues' arguments and you understand what a tonne of CO2 means.
- You don't really pay any attention to energy usage.

Given the fact that the average cost of a home energy monitoring system is between £200 and £1000, would you ever consider purchasing one for yourself? Why or why not?

Do you understand the energy consumed by the different appliances in your home?

What type of feedback do you think will affect the amount of energy you use?

Suggestions

- Receiving alerts when energy use exceeds a certain target
- Setting energy goals
- Comparing energy use against our past energy use
- Being able to compare against similar households
- Comparing energy usage against the average across all households in the neighbourhood

The image shows a mobile interface of an energy consumption monitoring app in two different scenarios; energy consumption targets achieved/not achieved. Do you like the idea of using such an interface to encourage users to save energy? (Ask for suggestions)

Do you have any questions?

Appendix B **Post-trial Interview Questions**

Mobile Application Usability

1. **How easily have you been able to navigate the application?**
2. **How well did you understand the information in the application?**
3. **Overall this application was easy to use**
4. **When using the mobile app "GreenAPP", Could you obtain information to answer any queries you had regarding your energy consumption?**
5. **Is the information displayed by the app timely and useful to you?**
6. **Are there tasks that you will want to perform that are not currently supported by application?**
7. **Is the application performance fast enough for you?**
8. **Are the graphics used recognizable to you and do they facilitate ease of use/understanding?**
9. **Can you successfully navigate through the application? Is the system status clear to you at all times? Is the navigation intuitive?**
10. **How do you rate the general handling of the app?**
11. **What do you think of the colour scheme?**

Energy Consumption Feedback

1. **When using GreenAPP, you energy consumption is feedback in power, monetary and CO2. Which is you preferred method?**
2. **Do you have any suggestion about the history feedback?**
3. **Do you have any other suggestion regarding the type of information provided by GreenAPP**

Appliance Energy Consumption

1. **How did you feel about being able to monitor you appliance consumption?**
2. **Do you think this experience will affect your future energy consumption behaviour?**
3. **Any other comments/ suggestions**

Overall Experience

1. **Did you enjoy the experience?**
2. **Would you invest in an energy monitor?**
3. **Any comments/suggestions**

Appendix C Trial/Interviews Consent Form

Information/Consent for Experiments/Interviews Participants

Project Name: Electricity Consumption Monitoring System
Researcher: Somoud Saqfelhait (MSc WAD Student at Leeds Metropolitan University)
Supervisor: Dr Muthu Ramachandran

Purpose of the Project:

Develop a mobile app to monitor and display information about the different appliances and the overall household electricity usage. We will be investigating if mobile phones can be used to help change people electricity consumption behaviour through constantly informing them about cost and carbon footprint of their electricity usage.

Potential Benefits

You will not benefit directly from participating in this study. However, you will be helping us to see the effect of the continuous awareness of electricity usage on the overall consumption of electricity which will have a positive environmental and financial benefit for the society.
The mobile app you will be testing will have the capability of displaying simple screen savers to reward green users and encourage energy wasters to do more.

Statement of confidentiality:

Any information that is obtained in connection with this study and that can be identified with you will remain confidential and will be disclosed only with your permission or as required by law. Confidentiality will be maintained by means of a code number to let the researcher know who you are. We will not use your name in any of the information we get from this study or in any of the research reports. When the study is finished, we will destroy the list that shows which code number goes with your name.

Information that can identify you individually will not be released to anyone outside the study. Ms. Saqfelhait will, however, use the information collected in her dissertation and other publications.

What is required of you?

1. provide two meter readings of your household with a week gap
2. attend a 20-40 min interview with the researcher who will ask you some questions regarding your household electricity consumption
3. The researcher will install a non-intrusive, safe and tested electricity monitoring system in your home. You could do that yourself if you prefer.
4. The system will send you electricity reading to a secure remote server which only you and the researcher can access

5. You will be provided with the mobile app which will enable you to view your energy consumption on the move. You will need an internet connection to be able to use the app effectively
6. The experiment will last 1 week during which you will have access to the researcher for any questions or problems
7. You will be required to attend a second interview with the researcher. This will last 15-20 min during which you will be asked about the effect of using the system on your household electricity consumption and whether you feel the experiment has affected your electricity consumption behaviour
8. Finally, you will be asked to provide 2 more readings of your electricity meter with a week gap.

Your right to withdraw

Your participation in this study is completely voluntary. The experiment and the subsequent interviews will be conducted in the period between 15/06/2011 – 30/07/2011

You have the right to withdraw from this experiment up to 15th August 2011.

Additional information

For more information please contact
Somoud Saqfelhait
s.saqfelhait3559@student.leedsmet.co.uk

Consent

I have had the opportunity to read this consent form, ask questions about the research project and I am prepared to participate in this project.

_____ _____

Participant's Signature Date

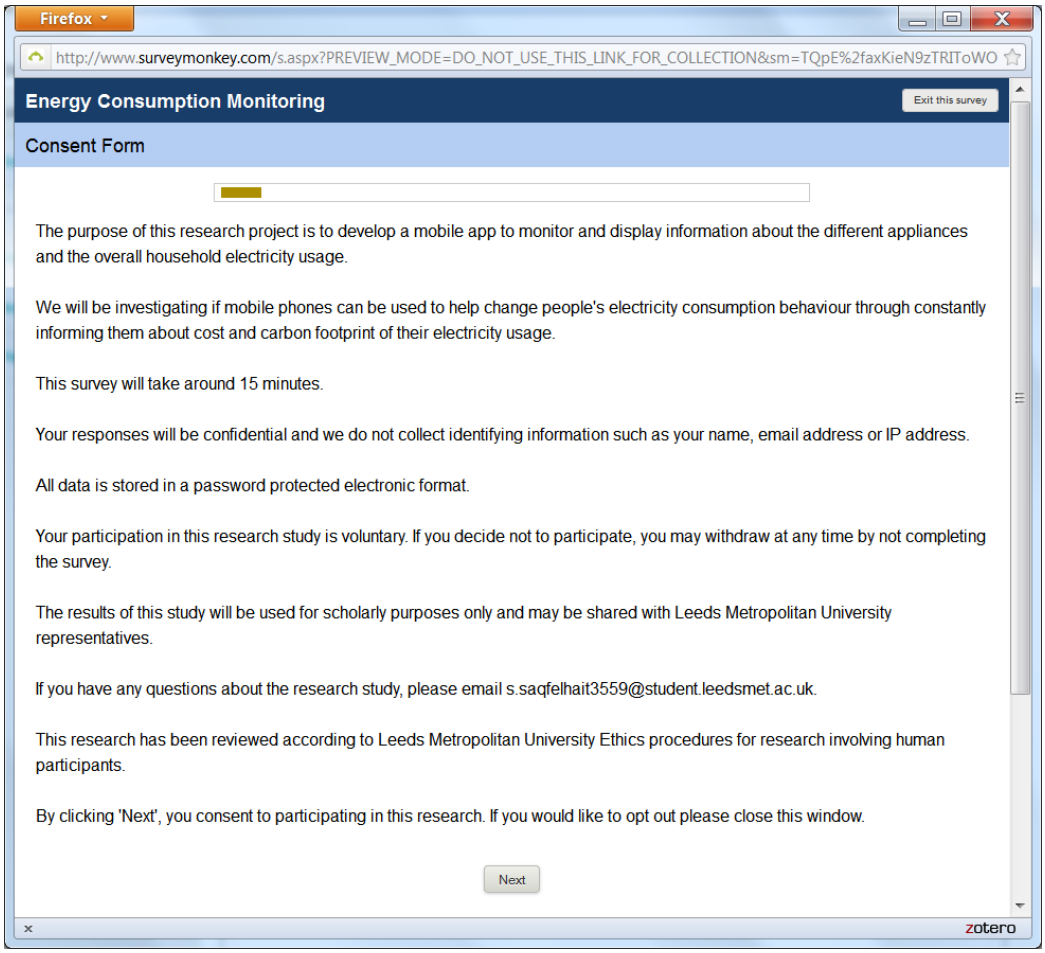
_____

Participant's Name

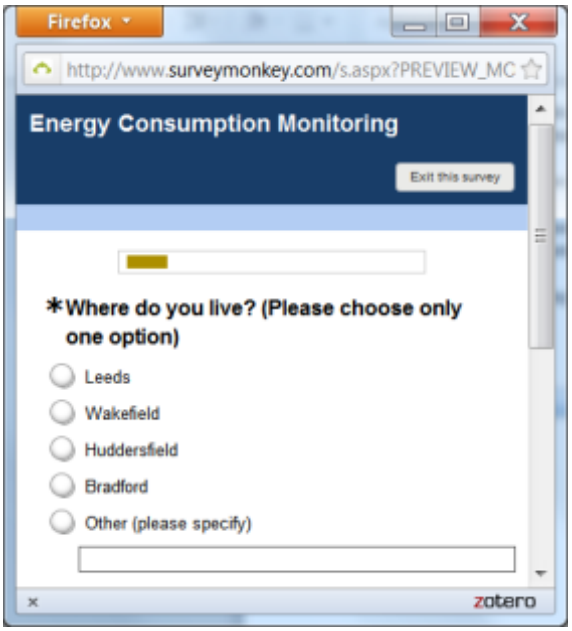
_____ _____

Researcher's Signature Date

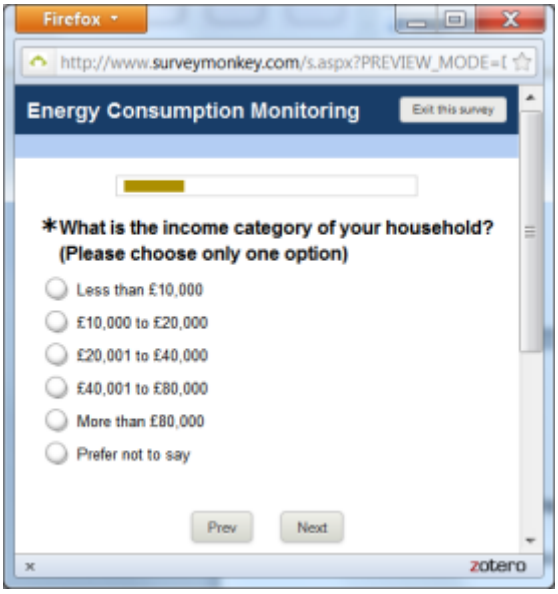
Appendix D **Online Survey questions**



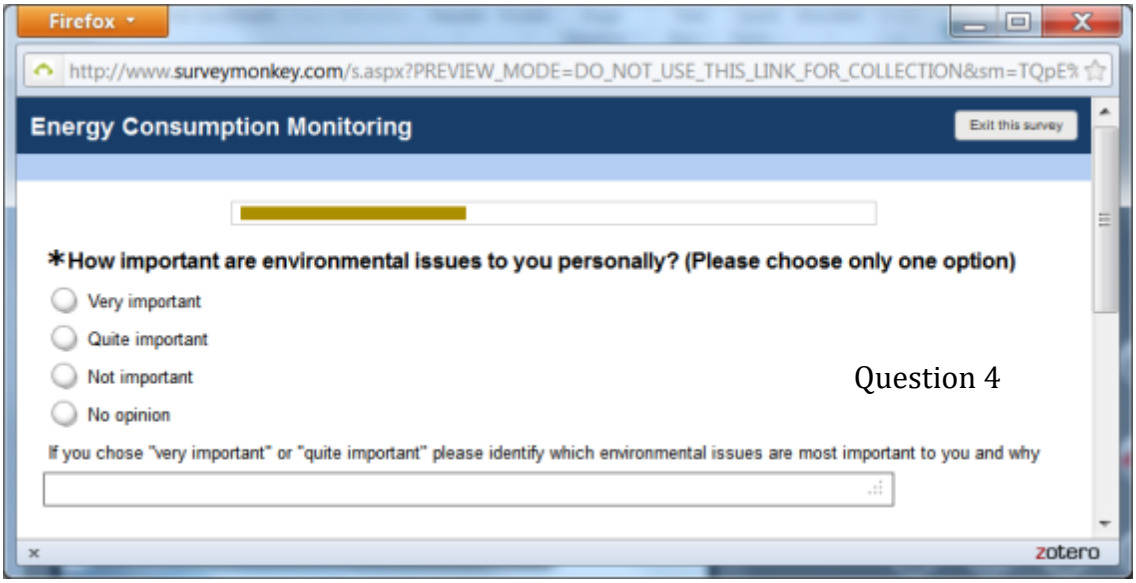
Question 1



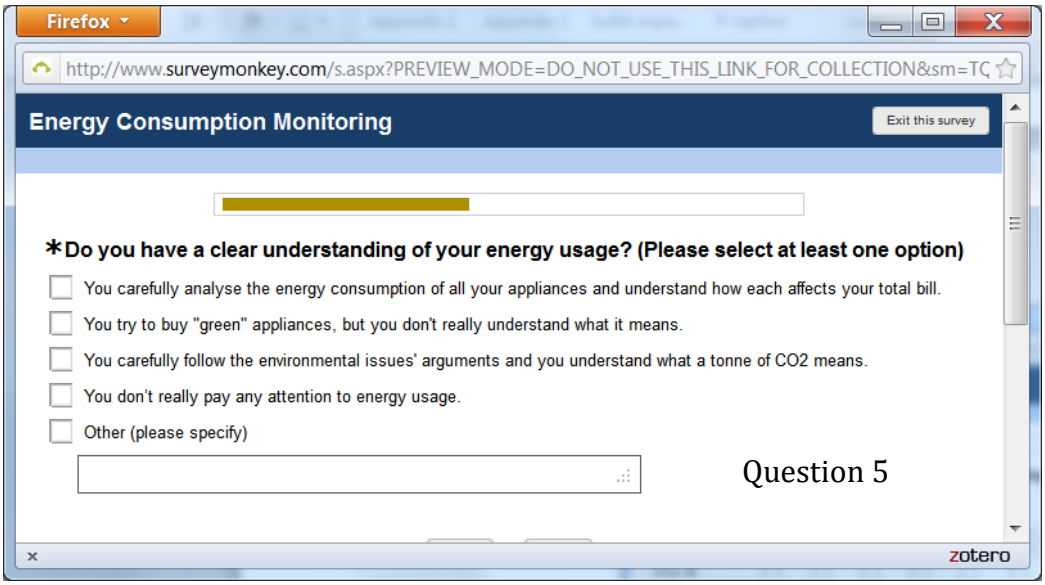
Question 2



Question 3

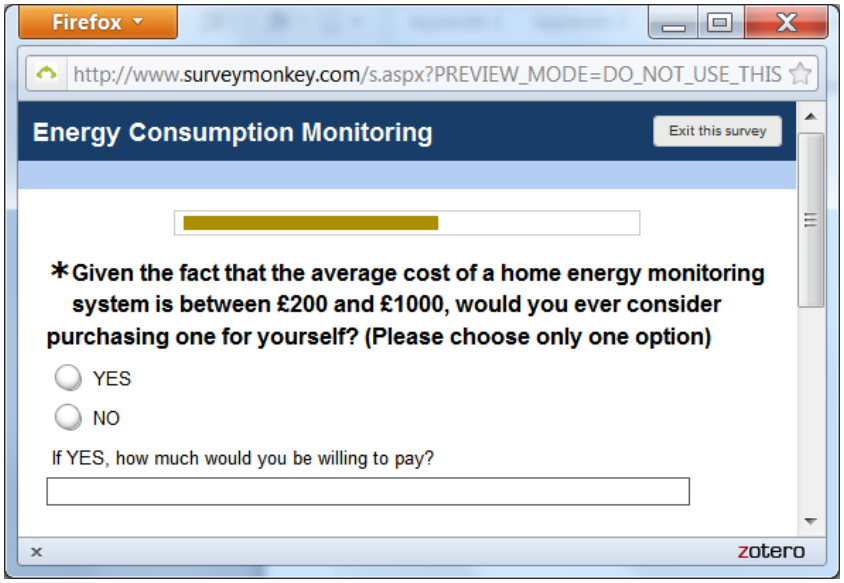


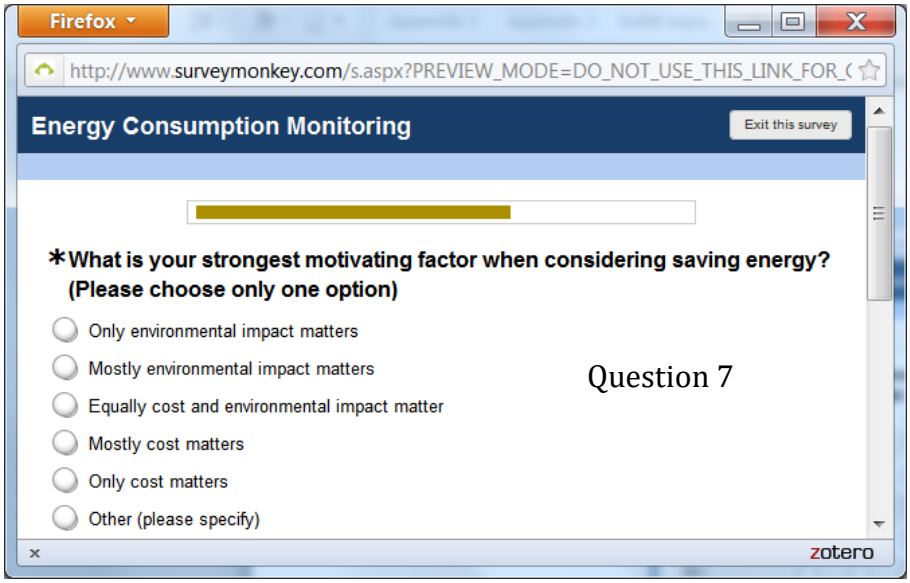
Question 4



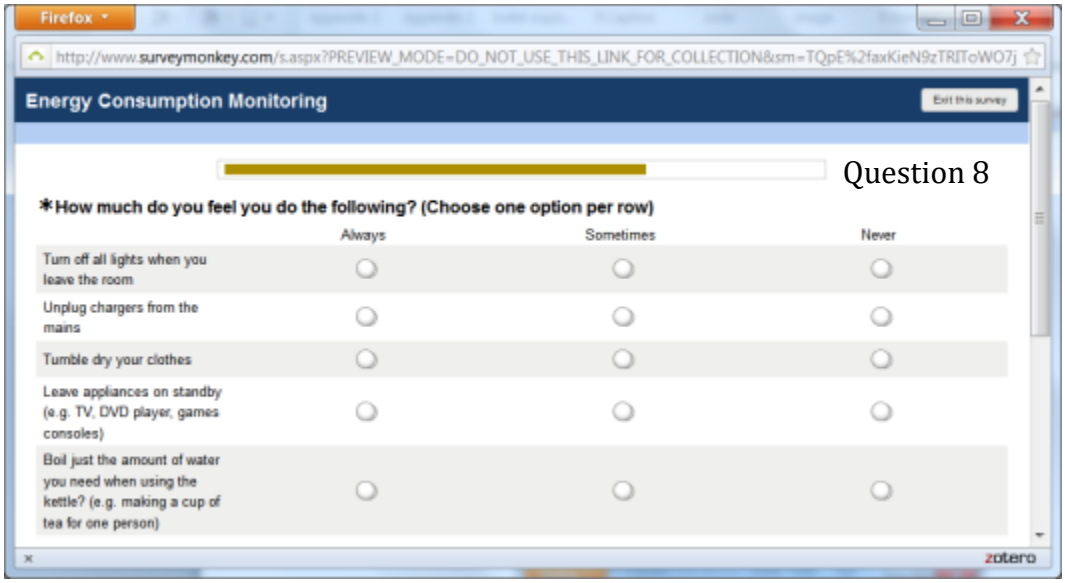
Question 5

Question 6



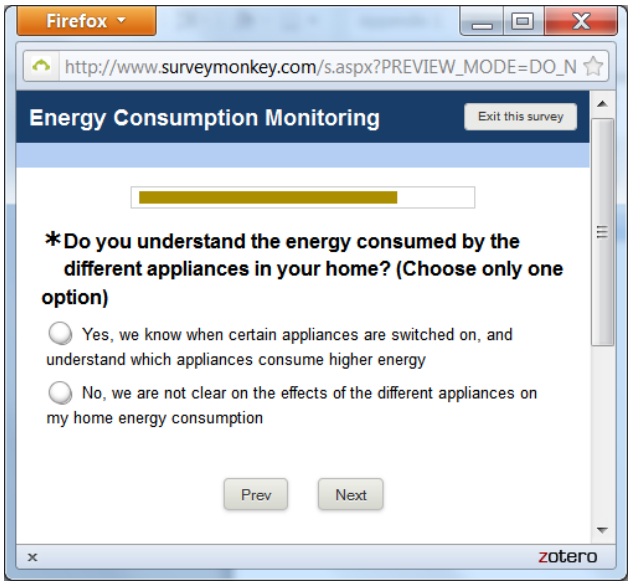


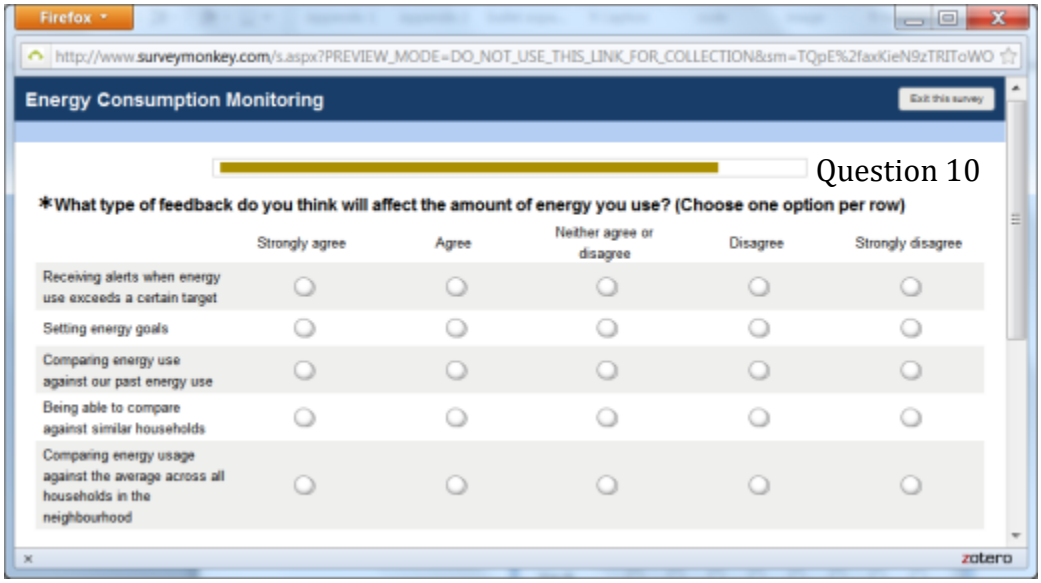
Question 7



Question 8

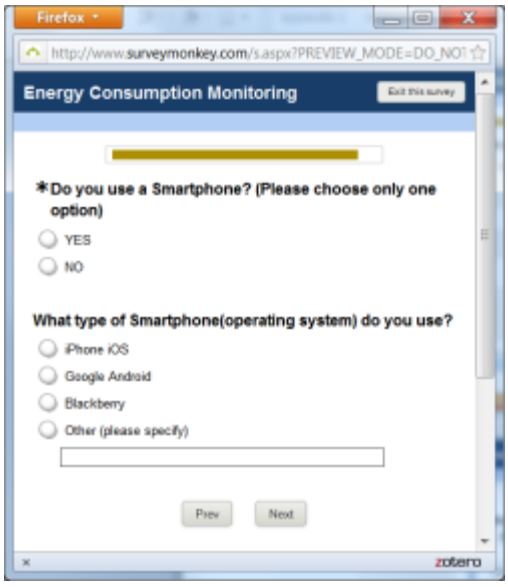
Question 9





Question 10

Question 11



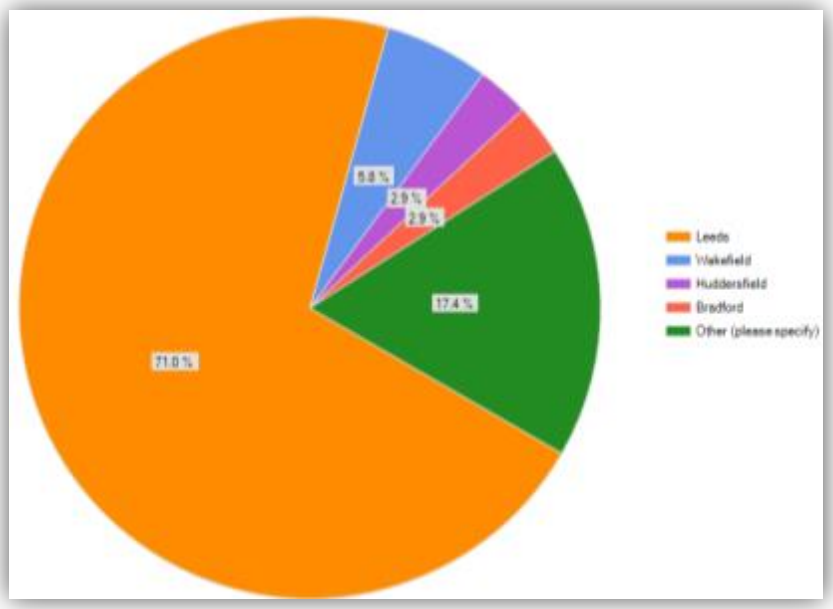
Question 12


Energy Consumption Monitoring
* The above image shows a mobile interface of an energy consumption monitoring app in two different scenarios;energy consumption targets achieved/not achieved. Do you like the idea of using such an interface to encourage users to save energy?
YES
NO
Please add any suggestions/comments about the interface

Appendix E **Online Survey Graphs and Tables**

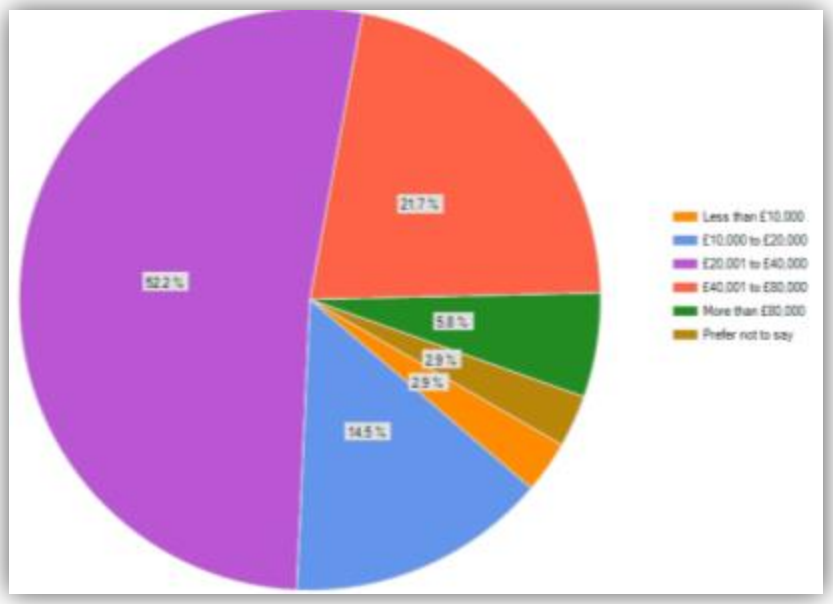
Participants classified according to their place of residence

Most of the participants who filled out the survey stated that they lived in the city of Leeds.

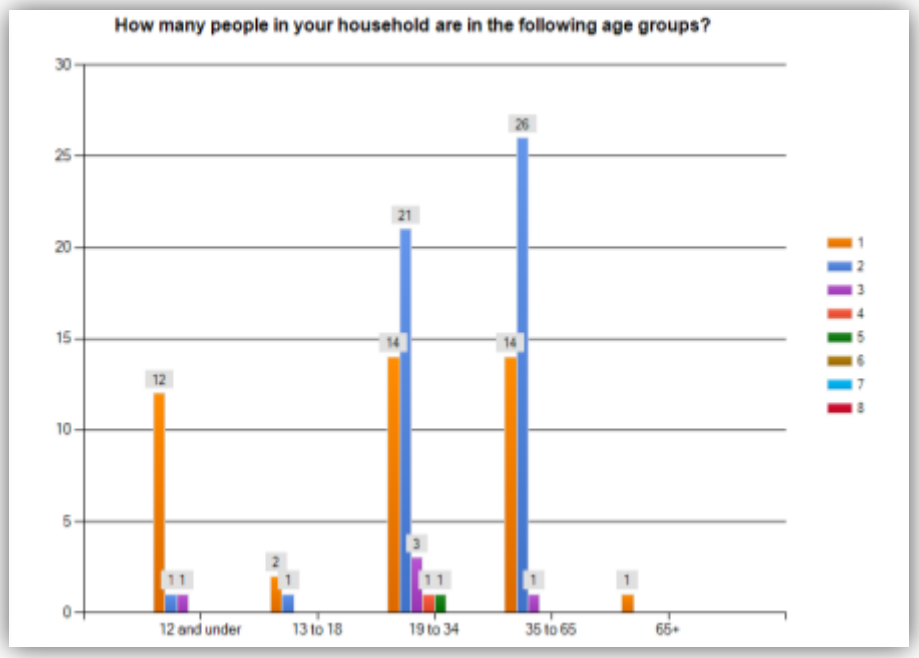


Participants classified according to household income

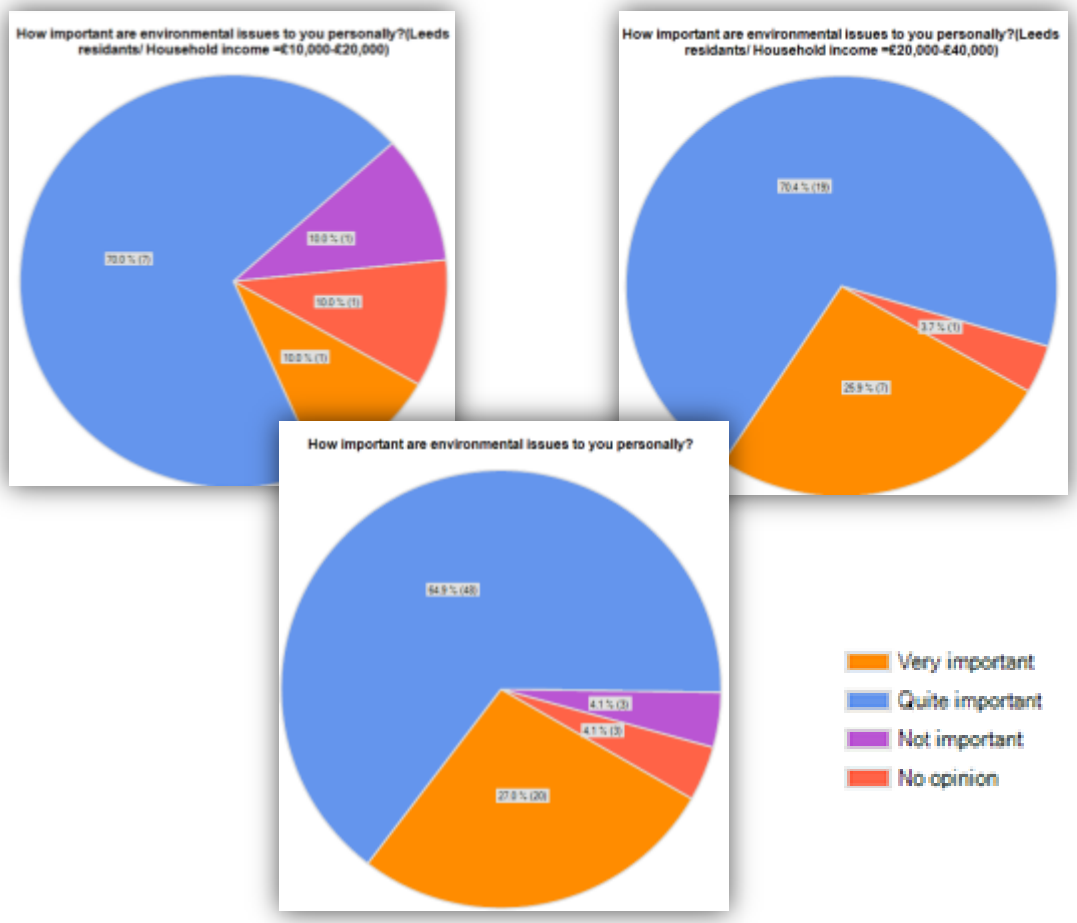
Over half of the participants had a household income of £20,000 to £40,000.



Participants' households grouped by age categories present

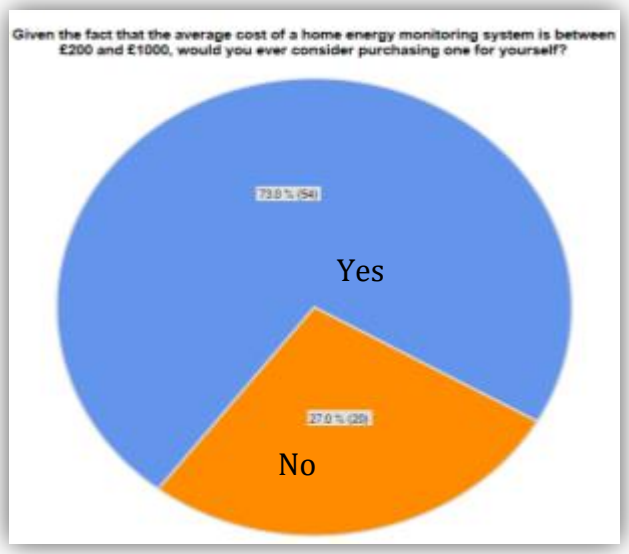


Level of importance of environmental issues to participants



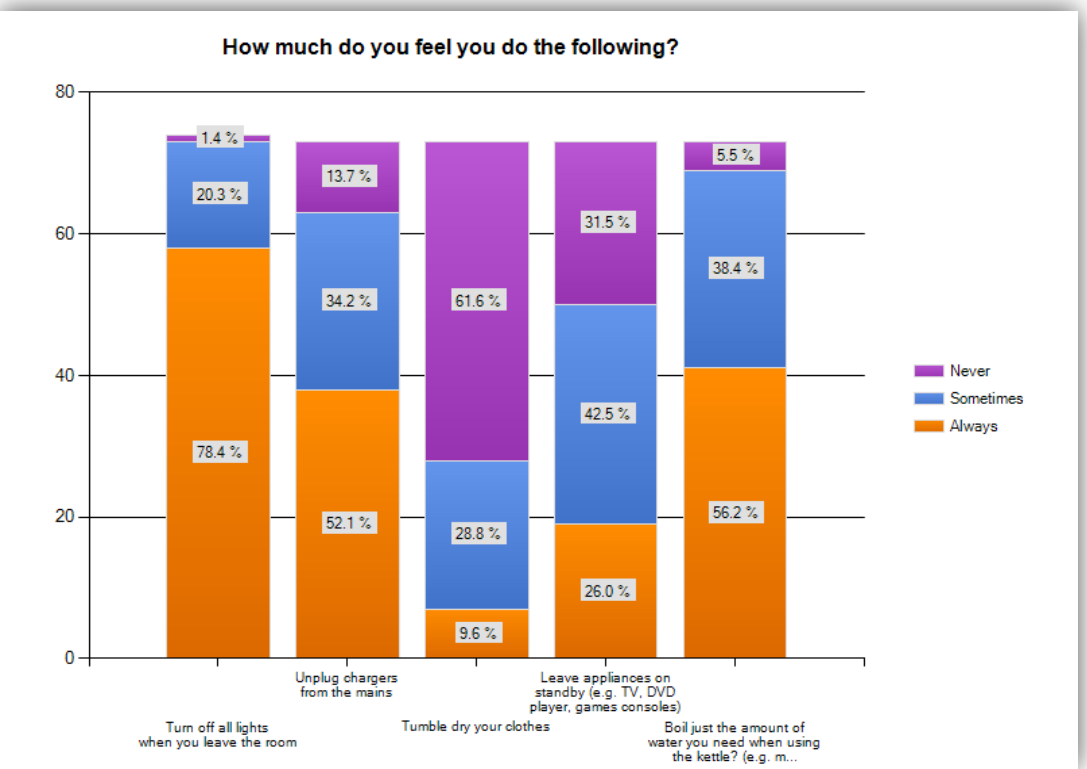
Respondent's willingness to invest in an energy monitoring system

Most of the participants said they wouldn't buy an energy monitor for the price range indicated.



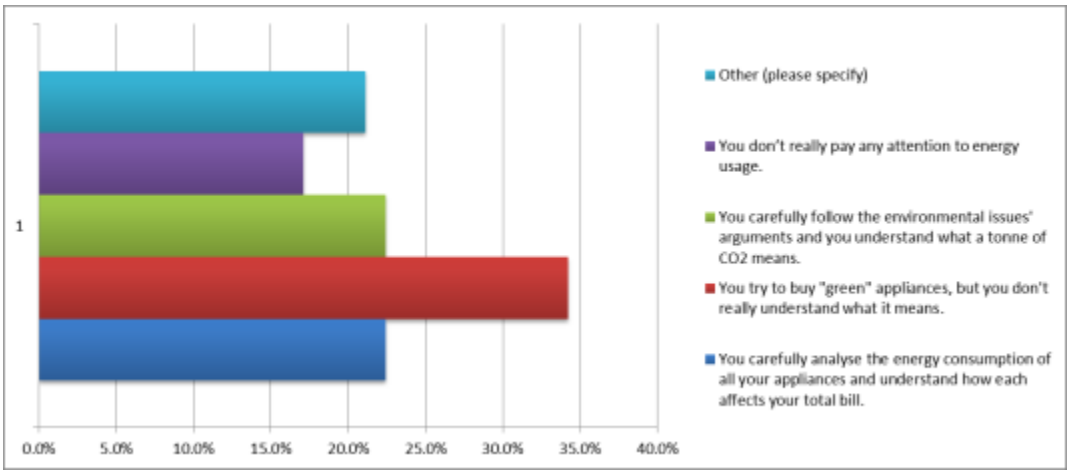
Energy consumption behaviour

The graph below shows the responses of participants to five different energy consumption actions, for example, 78.4% of respondents said they always turn off the light when they leave the room, 20.3% said they sometimes do it, and the remaining 1.4% said they never carry out this action.

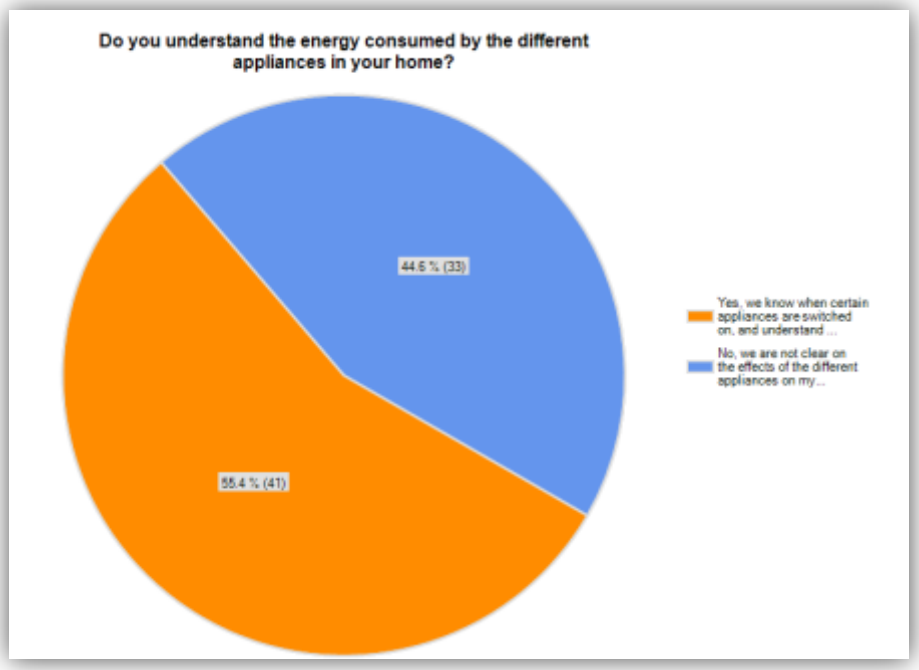


Understanding of energy and environment related terms

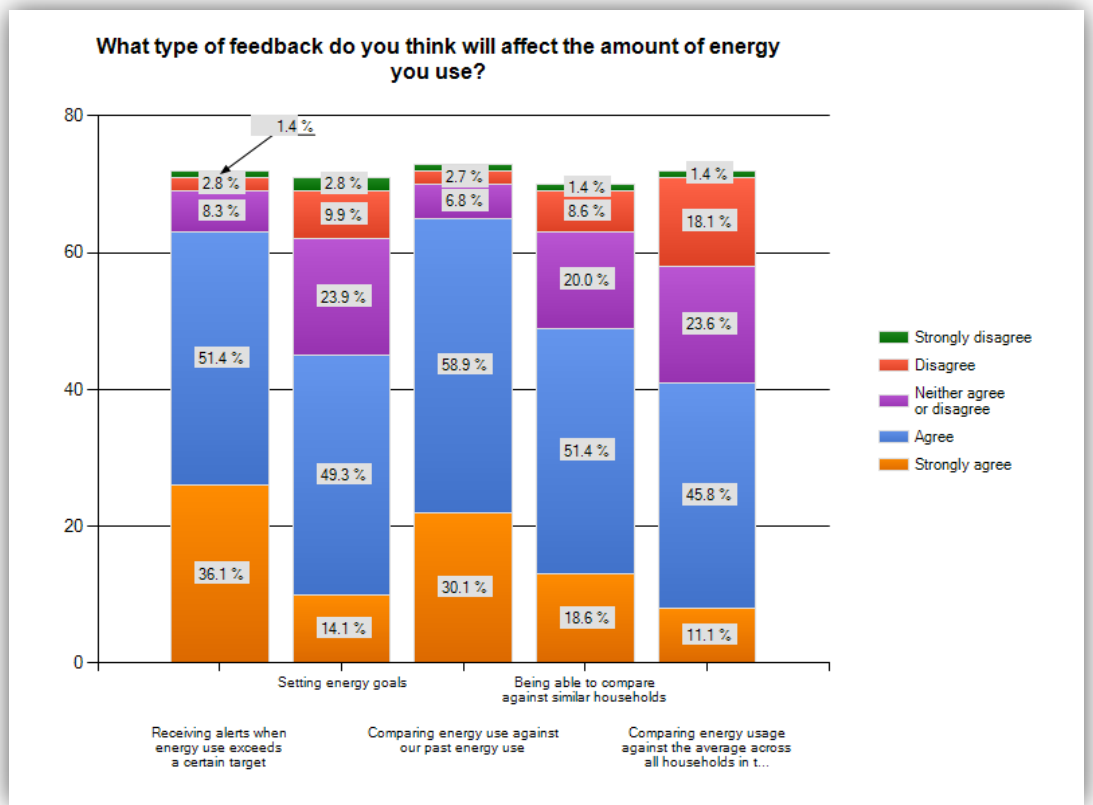
33.8 % of the participants felt they try to buy 'green' appliances but they don't really understand what it means. 17.6 % said they don't pay any attention to energy usage.



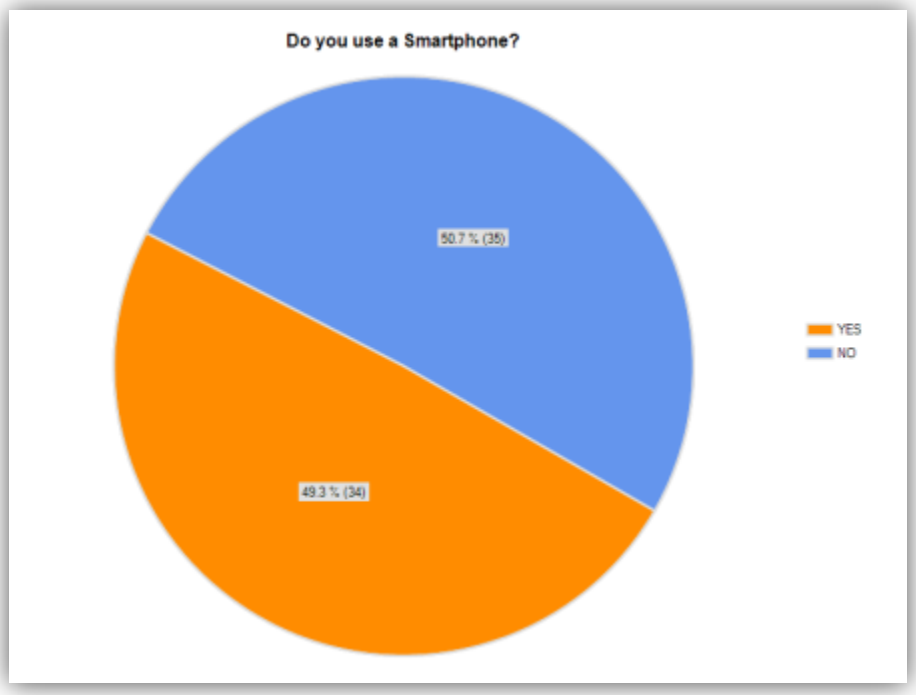
Respondents understanding of appliance energy consumption

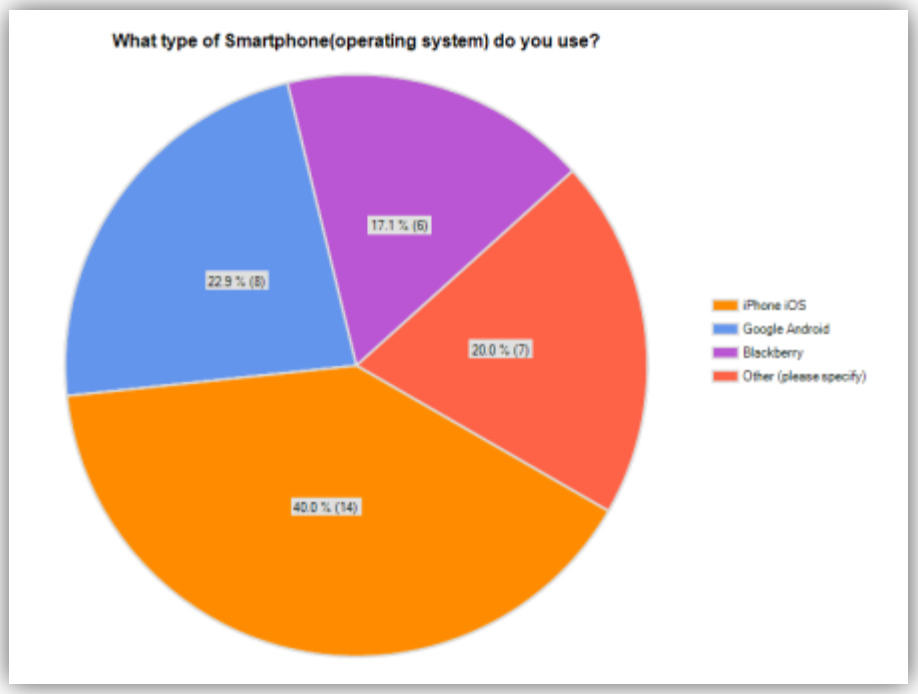


Energy Feedback influence on energy saving



Smart Phones and their Operating systems





Appendix F **Extracts of GreenAPP Implementation Source Code**

Main activity's code

```
package com.ide.green;

import java.io.IOException;
import java.text.DecimalFormat;
import java.util.Arrays;
import java.util.Date;

import android.app.Activity;
import android.content.BroadcastReceiver;
import android.content.ContentResolver;
import android.content.ContentValues;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.database.Cursor;
import android.database.SQLException;
import android.graphics.Color;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;
import android.widget.ViewSwitcher;

import com.androidplot.Plot;
import com.androidplot.xy.BarFormatter;
import com.androidplot.xy.BarRenderer;
import com.androidplot.xy.BoundaryMode;
import com.androidplot.xy.LineAndPointFormatter;
import com.androidplot.xy.LineAndPointRenderer;
import com.androidplot.xy.SimpleXYSeries;
import com.androidplot.xy.XYPlot;
import com.androidplot.xy.XYStepMode;

public class Main extends Activity {
/**
 * @author Somoud
 *
 */
public class DataReceiver extends BroadcastReceiver {
    /*
     * (non-Javadoc)
     *
     * @see
     * android.content.BroadcastReceiver#onReceive(android.content.Context,
     * android.content.Intent)
     */
    @Override
    public void onReceive(Context context, Intent intent) {

        updateSeries(intent);

    }
```

```
    /**
     * @param intent
     */
    private void updateSeries(Intent intent) {
        long time = 0;
        double power = 0;
        double temp = 0;
        if (intent.hasExtra("date"))
            time = intent.getExtras().getLong("date");
        if (intent.hasExtra("value1"))
            power = intent.getExtras().getDouble("value1");
        if (intent.hasExtra("value1"))
            temp = intent.getExtras().getDouble("value0");
        final TextView txt = (TextView) findViewById(R.id.txtPower);

        final TextView txtTemp = (TextView) findViewById(R.id.txtTemp);
        txtTemp.setText(String.valueOf(temp) + "C");
        final TextView txtCon = (TextView) findViewById(R.id.txtCon);
        final TextView txtCo2 = (TextView) findViewById(R.id.txtCo2);
        final TextView txtCost = (TextView) findViewById(R.id.txtCost);
        if (time > 0 && power > 0) {
            if (series2 == null) {
                series2 = new SimpleXYSeries("power");
            }

            series2.addLast(time, power);
            Date d = new Date();
            d.setHours(d.getTimezoneOffset());
            // get date at midnight today YYYY-MM-dd00:00:00
            d.setHours(0);
            d.setMinutes(0);
            d.setSeconds(0);
            // read the time of the last entry in the series
            long ti = 0;
            if (series2.size() > 1)
                ti = series2.getX(series2.size() - 2).longValue();
            else
                ti = time - (intUpdate * 60 * 1000);
            double sum = 0;
            try {
                sum = Commons.addKWH(power, ti, time);
            } catch (Exception e) {
                e.printStackTrace();
            }
            Log.i(intent.getDataString(),
                    "dataRecieved:w=" + String.valueOf(power) + ", time="
                            + String.valueOf(time) + ", computed energy="
                            + String.valueOf(sum));
            KWHenergy = KWHenergy + sum;
            DecimalFormat df = new DecimalFormat("#.##");
            txt.setText(df.format(power) + " " + dataType);
            txtCon.setText(df.format(KWHenergy) + " KWh");
            txtCo2.setText(df.format(KWHenergy * convertionFactor)
                    + " KgCo2e");
            txtCost.setText("" + df.format(KWHenergy * KWHprice));
            plot();
        }
    }
}

/**
```

```
 * @author Somoud
 *
 */
public class HistoryReceiver extends BroadcastReceiver {
    /*
     * (non-Javadoc)
     *
     * @see
     * android.content.BroadcastReceiver#onReceive(android.content.Context,
     * android.content.Intent)
     */
    @Override
    public void onReceive(Context context, Intent intent) {

        /*
         * initialise the data receiver to load data when a NEW_DATA_FOUND
         * broadcast is received
         */
        if (intent.hasExtra("msg")) {
            Commons.displayAlert(intent.getExtras().getString("msg"),
                    Main.this);
            unregisterReceiver(historyReceiver);
            stopHistoryService();
        } else {

            IntentFilter filter;
            try {

                filter = new IntentFilter(DataService.NEW_DATA_FOUND);
                dataReceiver = new DataReceiver();

                startDataService();
                registerReceiver(dataReceiver, filter);
            } catch (Exception e) {
                e.printStackTrace();
            }
            unregisterReceiver(historyReceiver);
            stopHistoryService();
            Log.i(intent.getDataString(), "historyRecieved");
            try {
                loadDataFromProvider();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
        ViewSwitcher switcher = (ViewSwitcher) findViewById(R.id.swtch);
        switcher.showNext();
    }
}

/**
 * conversion factor form KWh to kgCO2e
 */
final static double convertionFactor = 0.54522;
/**
 * font size for plot axes labels
 */
private static final float FONT_LABEL_SIZE = 16;
/**
 * Receiver of broadcasts from DataService
 */
```

```
private DataReceiver dataReceiver;
/**
 * stores the display data type can be W or KW
 */
private String dataType = "W";
/**
 * stores the graph type can be line or bar
 */
private String graphType = "line";
/**
 * Receiver of broadcasts from HistoryService
 */
private HistoryReceiver historyReceiver;

/**
 * stores the domain length of the graph
 */
private int intHours = 6;

// stores the energy for the day
private double KWHenergy = 0;
/**
 * stores update frequency
 */
private int intUpdate = 2;
/**
 * stores the energy price in pence
 */
private double KWHprice = .14;
/**
 * the View represnting the graph
 */
private XYPlot plt;

/**
 * stores all power data
 */
private SimpleXYSeries series2;

/**
 * the format of line or bar of the graph
 */
private LineAndPointFormatter series2Format;

// perform aesthetic operations on the graph view
private void formatplt() {
    int color = getResources().getColor(R.color.textColor);
    int gridColor = getResources().getColor(R.color.gridColor);
    int bg = getResources().getColor(R.color.bgColor);
    int alph = 200;
    int str = 2;
    plt.setBorderStyle(Plot.BorderStyle.NONE, null, null);
    plt.getGraphWidget().getBackgroundPaint().setColor(bg);

    // plt.getGraphWidget().getDomainOriginLinePaint().setAlpha(50);
    // plt.getGraphWidget().getRangeOriginLinePaint().setAlpha(50);

    plt.getGraphWidget().getGridBackgroundPaint().setAlpha(50);
    plt.getLayoutManager().remove(plt.getLegendWidget());

    plt.getGraphWidget().getGridLinePaint().setColor(gridColor);
```

```
        plt.getGraphWidget().setDomainLabelTickExtension(10);
        plt.getGraphWidget().setRangeLabelTickExtension(10);
        plt.getGraphWidget().getGridLinePaint().setAlpha(100);


    plt.getGraphWidget().getDomainLabelPaint().setTextSize(FONT_LABEL_SIZE);
        plt.getGraphWidget().getDomainLabelPaint().setStrokeWidth(str);
        plt.getGraphWidget().getDomainLabelPaint().setColor(color);
        plt.getGraphWidget().getDomainLabelPaint().setAlpha(alph);

        plt.getGraphWidget().getDomainOriginLabelPaint()
                .setTextSize(FONT_LABEL_SIZE);
        plt.getGraphWidget().getDomainOriginLabelPaint().setColor(color);
        plt.getGraphWidget().getDomainOriginLabelPaint().setAlpha(alph);
        plt.getGraphWidget().getDomainOriginLabelPaint().setStrokeWidth(str);

        plt.getGraphWidget().getRangeLabelPaint().setTextSize(FONT_LABEL_SIZE);
        plt.getGraphWidget().getRangeLabelPaint().setColor(color);
        plt.getGraphWidget().getRangeLabelPaint().setAlpha(alph);
        plt.getGraphWidget().getRangeLabelPaint().setStrokeWidth(str);

        plt.getGraphWidget().getRangeOriginLabelPaint()
                .setTextSize(FONT_LABEL_SIZE);
        plt.getGraphWidget().getRangeOriginLabelPaint().setColor(color);
        plt.getGraphWidget().getRangeOriginLabelPaint().setAlpha(alph);
        plt.getGraphWidget().getRangeOriginLabelPaint().setStrokeWidth(str);

        plt.getGraphWidget().setMargins(20, 20, 30, 20);

        plt.getGraphWidget().setRangeLabelWidth(50);
        // Customise domain/range labels

        plt.setRangeLabel("");
        plt.getDomainLabelWidget().getLabelPaint().setColor(color);
        plt.getDomainLabelWidget().getLabelPaint().setAlpha(alph);
        plt.getDomainLabelWidget().setMarginTop(2);
        plt.setDomainLabel("Graph's Domain: " + intHours + " Hours "
                + "                  Graph is updated every " + intUpdate
                + " Minutes");

        /*
         * Create a formatter to use for drawing a series using
         * LineAndPointRenderer:
         */
        plt.setDomainValueFormat(new CDateFormat());
        /*
         * Create a formatter to use for drawing a series using
         * LineAndPointRenderer:
         */
        series2Format = new LineAndPointFormatter(Color.rgb(200, 0, 0), // line
                // colour
                null, // point colour
                null); // fill colour

        // change line thickness
        series2Format.getLinePaint().setStrokeWidth(2);

    }

    /**
     * gets a specific column from the SQLite database through the content
```

```
     * provider
     *
     * @param col
     * @param factor
     * @return
     * @throws SQLException
     */
    private Number[] getCol(int col, Date d) throws SQLException {
        Number[] array = null;
        // get the data for the required period only

        ContentResolver cr = getContentResolver();
        String s = DataProvider.KEY_DATE + " >= ?";
        String[] sa;
        sa = new String[1];
        sa[0] = String.valueOf(d.getTime());
        Cursor c = cr.query(DataProvider.CONTENT_URI, null, s, sa, null);
        startManagingCursor(c);
        if (c != null) {
            if (c.getCount() > 0) {
                array = new Number[c.getCount()];

                int cnt = 0;

                if (c.moveToFirst()) {
                    do {
                        array[cnt] = c.getDouble(col);
                        cnt = cnt + 1;
                    } while (c.moveToNext());

                }
            }
            c.close();
        }
        return array;
    }

    /**
     * gets the most recent value stored in a specific column of the database
     *
     * @param col
     * @return
     * @throws SQLException
     */
    private double getVal(int col) throws SQLException {
        double value = 0;
        ContentResolver cr = getContentResolver();
        Cursor c = cr.query(DataProvider.CONTENT_URI, null, null, null, null);
        startManagingCursor(c);
        if (c != null) {
            if (c.getCount() > 0) {
                c.moveToLast();
                value = c.getDouble(col);
            }
            c.close();
        }
        c = null;
        return value;
    }

    /**
```

```
 * reads all data and its time stamps and update the text views
 */
private void loadDataFromProvider() {

    Number[] timeStamp = null;

    Number[] series2Numbers = null;

    double temperature = 0;
    final TextView txt = (TextView) findViewById(R.id.txtPower);
    final TextView txtTemp = (TextView) findViewById(R.id.txtTemp);
    final TextView txtCon = (TextView) findViewById(R.id.txtCon);
    final TextView txtCo2 = (TextView) findViewById(R.id.txtCo2);
    final TextView txtCost = (TextView) findViewById(R.id.txtCost);

    // get rid of decimal points in our range labels:

    try {
        Date d = new Date();
        d.setHours(0 + d.getTimezoneOffset());
        series2Numbers = getCol(DataProvider.COLUMN_POWER, d);
        timeStamp = getCol(DataProvider.COLUMN_DATE, d);
        temperature = getVal(DataProvider.COLUMN_TEMPERATURE);

        if ((series2Numbers != null) && (timeStamp != null)) {
            if (series2Numbers.length > 0 && timeStamp.length > 0) {
                try {
                    KWHenergy = Commons.computeKWH(series2Numbers,
                            timeStamp);

                    DecimalFormat df = new DecimalFormat("#.##");
                    txt.setText(df
                            .format(series2Numbers[series2Numbers.length -
1])
                            + " " + dataType);
                    txtTemp.setText(String.valueOf(temperature) + " C");
                    txtCon.setText(df.format(KWHenergy) + "KWh");
                    txtCo2.setText(df.format(KWHenergy * convertionFactor)
                            + " KgCo2e");
                    txtCost.setText("" + df.format(KWHenergy * KWHprice));
                    plt.removeSeries(series2);
                    series2 = new SimpleXYSeries(Arrays.asList(timeStamp),
                            Arrays.asList(series2Numbers), "Power");
                    plot();
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        }
    } catch (SQLException s) {
        s.printStackTrace();
    }

}

/** Called when the activity is first created. */
/*
 * (non-Javadoc)
 *
 * @see android.app.Activity#onCreate(android.os.Bundle)
 */
```

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    getWindow().setBackgroundDrawableResource(R.color.bgColor);

    // initialise the graph
    final TextView txtChannel = (TextView)
findViewById(R.id.txtChannelDescription);
    try {
        graphType = Commons.readFromFile("graphType", this);
    } catch (IOException e) {
        graphType = "line";
        e.printStackTrace();
    }

    try {
        dataType = Commons.readFromFile("dataType", this);
    } catch (IOException e) {
        dataType = "W";
        e.printStackTrace();
    }

    // loadDataFromProvider();
    try {
        intHours = Integer.parseInt(Commons.readFromFile("intHours",
this));
    } catch (IOException e) {
        intHours = 6;
    }
    try {
        intUpdate = Integer.parseInt(Commons
                .readFromFile("intUpdate", this));
    } catch (IOException e) {
        intUpdate = 2;
        e.printStackTrace();
    }
    try {
        KWHprice = Double.parseDouble(Commons.readFromFile("price", this))
/ 100;
    } catch (IOException e) {
        KWHprice = .14;
        e.printStackTrace();
    }
    // inflate the zoom control and catch the click event
    final Button btnGraph = (Button) this
            .findViewById(R.id.btnGraphSettings);

    btnGraph.setOnClickListener(new OnClickListener() {

        /*
         * (non-Javadoc)
         *
         * @see
android.view.View.OnClickListener#onClick(android.view.View)
         */
        @Override
        public void onClick(View arg0) {
            showGraphSettings();

        }
```

```
});
plt = (XYPlot) findViewById(R.id.plt);
/*
 * by default, AndroidPlot displays developer guides to aid in laying
 * out the plot. To get rid of them call disableAllMarkup():
 */
plt.disableAllMarkup();
formatplt();
// get channel
int i = 0;
try {
    i = Integer.parseInt(Commons.readFromFile("channel", this));
} catch (IOException e) {
    i = 1;
    try {
        Commons.writeToFile("1", "channel", this);
    } catch (IOException e1) {
        e1.printStackTrace();
    }
    e.printStackTrace();
}
String s = DataProvider.KEY_CHANNEL + " = ?";
String[] sa;
sa = new String[1];
sa[0] = String.valueOf(i);
// If the data is new, insert it into the provider.
Cursor c = getContentResolver().query(DataProvider.CONTENT_URI_APP,
        null, s, sa, null);
startManagingCursor(c);
if (c != null) {
    if (c.getCount() > 0) {
        c.moveToFirst();

        txtChannel
                .setText(c.getString(DataProvider.COLUMN_DESCRIPTION));

    } else if (c.getCount() == 0) {
        ContentValues values = new ContentValues();
        // add main monitor
        values.put(DataProvider.KEY_CHANNEL, 1);
        values.put(DataProvider.KEY_DESCRIPTION, "Electricity Meter");
        try {
            getContentResolver().insert(DataProvider.CONTENT_URI_APP,
                    values);
        } catch (SQLException e) {
            e.printStackTrace();
        }
        for (int j = 2; j <= 10; j++) {

            values.put(DataProvider.KEY_CHANNEL, j);
            values.put(DataProvider.KEY_DESCRIPTION, "Channel " + j);
            try {
                getContentResolver().insert(
                        DataProvider.CONTENT_URI_APP, values);
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }

    }
    c.close();
```

```
        }
        c = getContentResolver().query(DataProvider.CONTENT_URI, null, null,
                null, null);
        if (c == null) {
            IntentFilter filter;
            filter = new IntentFilter(HistoryService.HISTORY_LOADED);
            historyReceiver = new HistoryReceiver();
            startHistoryService();
            registerReceiver(historyReceiver, filter);
        } else if (c.getCount() == 0) {
            IntentFilter filter;
            filter = new IntentFilter(HistoryService.HISTORY_LOADED);
            historyReceiver = new HistoryReceiver();
            startHistoryService();
            registerReceiver(historyReceiver, filter);
            c.close();
        } else {
            IntentFilter filter;
            try {

                filter = new IntentFilter(DataService.NEW_DATA_FOUND);
                dataReceiver = new DataReceiver();

                startDataService();
                registerReceiver(dataReceiver, filter);
            } catch (Exception e) {
                e.printStackTrace();
            }
            try {

                loadDataFromProvider();
            } catch (SQLException e) {
                e.printStackTrace();
            }
            ViewSwitcher switcher = (ViewSwitcher) this
                    .findViewById(R.id.swtch);
            switcher.showNext();
            c.close();
        }
        c = null;
    }

    /*
     * (non-Javadoc)
     *
     * @see android.app.Activity#onCreateOptionsMenu(android.view.Menu)
     */
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        super.onCreateOptionsMenu(menu);
        // add the main menu items
        menu.add(0, 0, 0, "Select or Edit Appliance");
        menu.add(0, 1, 1, "Change Feed ID/API Key");
        menu.add(0, 2, 2, "Energy Prices");
        menu.add(0, 3, 3, "Detailed History");
        menu.add(0, 4, 4, "Exit");
        return true;
    }

    /** Called when the activity is first created. */
    /*
```

```
     * (non-Javadoc)
     *
     * @see android.app.Activity#onDestroy()
     */
    /*
     * (non-Javadoc)
     *
     * @see android.app.Activity#onDestroy()
     */
    @Override
    public void onDestroy() {

        try {
            unregisterReceiver(historyReceiver);
            stopHistoryService();
        } catch (IllegalArgumentException e) {
            e.printStackTrace();
        }
        // unload service
        try {
            unregisterReceiver(dataReceiver);
            stopDataService();
        } catch (IllegalArgumentException e) {
            e.printStackTrace();
        }
        super.onDestroy();
    }

    /*
     * (non-Javadoc)
     *
     * @see android.app.Activity#onOptionsItemSelected(android.view.MenuItem)
     */
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
        case 0:
            showAppliance();
            return true;
        case 1:
            showSettings();
            return true;
        case 2:
            showPrice();
            return true;
        case 3:
            showHistory();
            return true;
        case 4:
            finish();
            return true;
        }
        return true;
    }

    /**
     * draws a graph
     *
     */
    private void plot() {
```

```
        // calculate the max and min of the y value to format the y axis
        double max = Commons.getMax(intHours, series2, 2);
        double min = Commons.getMin(intHours, series2, 2);
        plt.setRangeBoundaries(min, max, BoundaryMode.AUTO);
        if (max > 0) {
            double step = (max - min) / 5.0;
            int nearest = (int) Math.round(Math.log10(max - min) / 2);
            double rounded = Commons.roundToNextTen(step, nearest);
            if (rounded < 1)
                rounded = 1; // make sure the step is not zero
            plt.setRangeStep(XYStepMode.INCREMENT_BY_VAL, rounded);
            plt.setTicksPerDomainLabel(4);
        }

        plt.setDomainStep(XYStepMode.INCREMENT_BY_VAL, intHours * 60 * 60
                * 1000 / 8);

        if (dataType.equals("KW")) {
            plt.setRangeValueFormat(new KwFormat("#.###KW"));
        } else {
            plt.setRangeValueFormat(new DecimalFormat("0" + dataType));
        }
        if (graphType.equals("bar")) {
            plt.addSeries(
                    series2,
                    BarRenderer.class,
                    new BarFormatter(Color.argb(100, 0, 200, 0), Color.rgb(0,
                            80, 0)));
        } else {
            plt.addSeries(series2, LineAndPointRenderer.class, series2Format);
        }
        plt.redraw();

        long maxTime = Commons.getMaxTime();
        if (maxTime > 0) {
            plt.setDomainBoundaries(maxTime - (intHours * 3600000) - 1800000,
                    maxTime, BoundaryMode.AUTO);
        } else {
            if (intHours == 2) {
                plt.setTicksPerDomainLabel(4);
            } else {
                plt.setTicksPerDomainLabel(2);
            }
        }
    }

    /**
     * launch ApplianceActivity
     */
    private void showAppliance() {
        Intent applianceActivity = new Intent(this, Appliance.class);
        startActivity(applianceActivity);
        finish();
    }

    /**
     * launch Graph Settings Activity
     */
    private void showGraphSettings() {
        Intent graphSettingsActivity = new Intent(this, GSettings.class);
        // If your target activity can handle many intentions you would
```

```
    // need to pass the code for the intention that you have
    startActivity(graphSettingsActivity);
    finish();
}

/**
 * launch Detailed History activity
 */
private void showHistory() {

    Intent activity = new Intent(this, DHistory.class);
    startActivity(activity);
    finish();
}

/**
 * launch price activity
 */
private void showPrice() {

    Intent activity = new Intent(this, Price.class);
    startActivity(activity);
    finish();
}

/**
 * launch Settings activity
 */
private void showSettings() {

    Intent settingsActivity = new Intent(this, Settings.class);
    startActivity(settingsActivity);
    finish();
}

/**
 * start services explicitly
 */

private void startDataService() {
    startService(new Intent(this, DataService.class));
}

/**
 * start services explicitly
 */

private void startHistoryService() {
    startService(new Intent(this, HistoryService.class));
}

/**
 * stop service explicitly
 */
private void stopDataService() {
    stopService(new Intent(this, DataService.class));
}

/**
 * stop service explicitly
 */
```

```
private void stopHistoryService() {
    stopService(new Intent(this, HistoryService.class));
}
}
```

Appendix G **Computer based GreenAPP Data Transmitter Source Code**

Main Program

```
import eeml.*;
import processing.serial.*;

// Augest 2011

Serial myPort;
int val;
String buffer = "";
String message = "";
int startPos;
int endPos;
XMLElement xml;
float[] watts;
float temperature = 0;
int [] noWatts;
int noTemperature=0;
DataOut dOut;
int updateInterval = 60; // seconds
double lastUpdate;
double lastMessage;
int numberOfChannels;
String debugMessage = "";
PFont font;
String remoteServerURL;
String remoteServerAPI = "";
int serialDevice;
boolean monitorFound = false;
String loadPath = "preferences.txt";
int historySize = 280;//holds the width of the history graph

int[][] historicValues;

String this_model;
String this_area;

boolean modelVerified = false;
boolean datastreamsSet = false;
int[] serial_speed = {
  57600, 9600
};

int this_serial_speed;
String unit = "";

void setup() {
  size(320, 500);//window size
  frameRate(2);
  watts = new float[10];
  noWatts = new int[10];
  font = loadFont("Univers-Bold-12.vlw");
  loadPreferences();
  this_model = "";
  this_area = "";

  if (!remoteServerAPI.equals("")) {

    try {
```

```
      this_serial_speed = 0;
      setupSerial(serial_speed[this_serial_speed]);
      setupRemoteServer();
      fill(0, 255, 0);
      debugMessage = "\nNo info received yet\nDevice selected: \n"
        + Serial.list()[serialDevice] + "\n\nChannels: "
        + numberOfChannels;
    }
    catch (Exception e) {
    }
  }
}

void draw() {

  background(255, 255, 255);

  while (myPort.available () > 0) {
    if (!modelVerified) {
    }
    String inBuffer = myPort.readString();
    if (inBuffer != null) {
      buffer += inBuffer;
      if (!modelVerified) {

        boolean nonsenseXML = false;

        for (int i = 0; i < buffer.length(); i++) {
          int b = (int) buffer.charAt(i);
          if (b > 255) {
            nonsenseXML = true;
          }
        }

        if (nonsenseXML) {
          this_serial_speed = (this_serial_speed + 1) % 2;
          myPort.clear();
          myPort.stop();
          setupSerial(serial_speed[this_serial_speed]);
          debugMessage = "Trying new serial port \nspeed: "
            + serial_speed[this_serial_speed]
            + ". Please wait\na few seconds";
          lastUpdate = millis();
        }
      }

      startPos = buffer.indexOf("<msg>");
      endPos = buffer.indexOf("</msg>");

      if ((startPos >= 0) && (endPos > 0)) {
        if (endPos > startPos) {
          message = buffer.substring(startPos, endPos);
          buffer = "";
          InputParse(message);
          monitorFound = true;
          modelVerified = true;
          lastMessage = millis();
        }
        else {
          buffer = buffer.substring(startPos, buffer.length());
        }
```

```
      }
    }
  }

  if ((millis() - lastUpdate) > updateInterval * 1000) {
    if (modelVerified) {
      sendData();
      lastUpdate = millis();
    }
  }

  if (millis() < lastUpdate)
    lastUpdate = millis();

  fill(0, 0, 0);

  textFont(font, 16);
  text(debugMessage, 30, 30);

  textFont(font, 12);
  text("Time to remote server update: "
    + (int) (updateInterval - (millis() - lastUpdate) / 1000)
    + " secs", 30, 275);
  fill(255, 255, 255);
  rect(-1, 280, width + 2, height - 280);

  fill(0, 0, 0);
  text("Serial Port", 10, 306);
  text("Feed URL", 10, 346);
  text("API KEY", 10, 386);
  text("Transmit frequency(Minutes)", 10, 426);
  drawGraph();
}
```

Extract power and temperature from the energy monitor XML data

```
//extract data

void InputParse(String input) {

  if ((input.indexOf("<hist") < 0) || (serial_speed[this_serial_speed] ==
9600)) {

    try {

      debugMessage = "GreenAPP Data Transmitter\n\nPower: \n";

      if (input.indexOf("<tmprF") < 0) {
        this_area = "UK";
      }
      else {
        this_area = "US";
      }
      int channel;
      channel=int(parseSingleElement(input, "sensor"));
      int Watt = int(parseSingleElement(input, "watts"));

      arrayCopy(historicValues[channel], 1, historicValues[channel], 0,
historySize-1);
      historicValues[channel][historySize-1] =Watt;//add value to graph
    //add to the data collector
```

```
    if(noWatts[channel]==0){
      watts[channel]=float(Watt);
      noWatts[channel]=noWatts[channel]+1;
    }
    else{

watts[channel]=Round(((noWatts[channel]*watts[channel])+Watt)/(noWatts[chan
nel]+1),1);
      noWatts[channel]=noWatts[channel]+1;
    }
 debugMessage += "Channel "+((channel+1) + " - " + watts[channel] + "
W\n");
      debugMessage += "\nTemperature: ";
   if(noTemperature==0){
       temperature = float(parseSingleElement(input, "tmpr"));
      noTemperature=noTemperature+1;
    }
    else {
      temperature=Round(((noTemperature*temperature)+
float(parseSingleElement(input, "tmpr")))/(noTemperature+1),1);
      noTemperature=noTemperature+1;
    }
        unit = "C";
      debugMessage += (temperature + " "+unit+"\n");

      if (!datastreamsSet) {
        setupDatastreams();
      }
    }

    catch (Exception e) {
    }
  }
}

String parseSingleElement(String input, String target) {
  int start = input.indexOf("<" + target + ">") + target.length() + 2;
  int end = input.indexOf("</" + target + ">");
  return (input.substring(start, end));
}
```

Send data to remote server

```
void sendData() {
  if (monitorFound) {

    dOut.update(0, temperature);
    temperature=0;//initialise temoerature collector
    noTemperature=0;
    for (int i = 0; i < 9; i++) {
      if (watts[i] >= 0){
        dOut.update(i + 1, watts[i]);
        watts[i]=0;//initialise the collector
        noWatts[i]=0;
      }
    }

    int response = dOut.updatePachube();
    if (response == 200) {
```

```
      debugMessage += "\n** updated remote server**";
    }
    else {

      debugMessage = "Problem updating remote server\n";

      if (response == 404)
        debugMessage += "\nFeed does not exist";
      if (response == 401)
        debugMessage += "\nYou don't own that feed";
      if (response == 503)
        debugMessage += "\nPachube server error";
    }
  }
  else {
    debugMessage = "\n** no remote server update **";
  }
}


void setupRemoteServer() {
  dOut = new DataOut(this, remoteServerURL, remoteServerAPI);
  lastUpdate = millis();
}
void resetTransmitter(){
  dOut.quit();
  dOut = new DataOut(this, remoteServerURL, remoteServerAPI);
  setupDatastreams();
}


void setupDatastreams() {
    dOut.addData(0, "temperature, degrees, celsius");
    dOut.setUnits(0, "Celsius", "C", "basicSI");
  for (int i = 0; i < numberOfChannels; i++) {
    watts[i] = 0;
    dOut.addData(i + 1, "watts, electricity, power");
    dOut.setUnits(i + 1, "Watts", "W", "derivedSI");
  }

  datastreamsSet = true;}
```

Setup Connection to the energy monitor via serial port

```
void setupSerial(int serial_speed) {
  try {
    String portName = Serial.list()[serialDevice];
    myPort = new Serial(this, portName, serial_speed);
    debugMessage = "\nNo info received yet\nDevice selected: \n"
      + Serial.list()[serialDevice] + "\n\nChannels: "
      + numberOfChannels;
  }
  catch (Exception e) {
  }
  lastMessage = millis();
  buffer = "";
}
```

Plot the data stored in the history array

```
void drawGraph() {
```

```
  fill(255, 255, 255);
  stroke(0);
  strokeWeight(1);

  rect(20, 175, 280, 85);

  stroke(230, 230, 230);

  for (int i = 1; i < 28; i++) {
    line(20 + i * 10, 259, 20 + i * 10, 176);
  }

  color from = color(255, 0, 255);
  color to = color(0, 255, 255);

  int[] maxWatts;
  maxWatts = new int[numberOfChannels];

  for (int i = 0; i < numberOfChannels - 1; i++) {
    maxWatts[i] = max(historicValues[i]);
  }

  int maxAllWatts = max(maxWatts);

  for (int i = 0; i < numberOfChannels; i++) {

     color graphLine = lerpColor(from, to, (float) i
        / (float) 2);
    stroke(graphLine);
    for (int j = 2; j < historySize; j++) {

        float graphHeight1 = 80.0 * (float) historicValues[i][j - 1]
          / (float) maxAllWatts;
        float graphHeight2 = 80.0 * (float) historicValues[i][j]
          / (float) maxAllWatts;
        line((float) 19 + j, height - 241 - graphHeight1, (float) 20
          + j, height - 241 - graphHeight2);
    }
  }
  stroke(0);
}
```

Round function: to round data to nearest one decimal place

```
//round val to nearest dec decimal places
public static float Round(float val, int dec) {
   float tens = (float)Math.pow(10,dec);
   val = val * tens;
   float tmp = Math.round(val);
   return (float)tmp/tens;
   }
```

GUI setup and mouse Event Listener

```
import SpringGUI.*;

SpringGUI gui;

void setupGUI() {
  int xposition = 71;
```

```
  int yposition = 4;
  gui = new SpringGUI(this);
  println(Serial.list());
  gui.addChoice("serial", 70 + yposition, 220 + xposition, 240, 20);
  for (int i = 0; i < Serial.list().length; i++) {
    gui.addItem("serial", Serial.list()[i]);
  }
  gui.selectItem("serial", Serial.list()[serialDevice]);
  gui.addTextField("url", remoteServerURL, 70 + yposition, 260 + xposition,
240, 25);
  gui.addTextField("api", remoteServerAPI, 70 + yposition, 300 + xposition,
240, 25);
  gui.addTextField("update",Integer.toString(updateInterval/60), 180 +
yposition, 340 + xposition, 110, 25);
  gui.addButton("save", "save preferences", 70 + yposition, 380 +
xposition, 150, 40);
}
void handleEvent(String[] parameters) {

  if ((parameters[1].equals("save")) &&
(parameters[2].equals("mouseClicked"))) {
    String s= gui.getText("serial");
    String c= gui.getText("sensors");

    for (int i = 0; i < gui.getItemCount("serial"); i++) {
      if (gui.getSelectedItem("serial").equals(gui.getItemByIndex("serial",
i))) {
        serialDevice = i;
      }
    }
    remoteServerURL= gui.getText("url");
    remoteServerAPI= gui.getText("api");
    updateInterval=int (gui.getText("update"))*60;
    println(serialDevice + " " +numberOfChannels + " "+ remoteServerURL + "
" +remoteServerAPI + " " +Integer.toString(updateInterval)+" ");
    savePrefs(Integer.toString(serialDevice),
Integer.toString(numberOfChannels), remoteServerURL,
remoteServerAPI,Integer.toString(updateInterval/60));
    resetTransmitter();
  }
}
```

Load/ Save preference functions

```
void savePrefs(String a, String b, String c, String d, String e) {
  try {
    String[] newSave;
    newSave = new String[5];
    newSave[0] = a;
    newSave[1] = b;
    newSave[2] = c;
    newSave[3] = d;
    newSave[4] = e;
    saveStrings(loadPath, newSave);

    serialDevice = int(a);
    numberOfChannels = 10;
    remoteServerURL = c;
    remoteServerAPI = d;
    updateInterval=int(e)*60;//seconds
```

```
    temperature=0;//initialise temoerature collector
    noTemperature=0;
    for (int i = 0; i < numberOfChannels; i++) {
       watts[i]=0;//initialise the collector
        noWatts[i]=0;
    }
  }
  catch (Exception e) {
  }
}
void loadPreferences() {

  try {
    String lines[] = loadStrings(loadPath);
    if (lines.length >= 4) {
      serialDevice = int(lines[0]);
      numberOfChannels= 10;
      remoteServerURL = lines[2];
      remoteServerAPI = lines[3];
      updateInterval = int(lines[4])*60;
    }
    else {
      savePrefs("0", "10", "url", "api", "5");
    }
    setupGUI();
    historicValues = new int[numberOfChannels][historySize];
    for (int i = 0; i < numberOfChannels; i++) {
      for (int j = 0; j< historySize; j++) {
        historicValues[i][j] = 0;
      }
    }
  }
  catch (Exception e) {
    savePrefs("0", "10", "url", "api", "5");
    fill(0);
    textFont(font, 18);
    text("Created new preferences\nfile... \nPlease restart app", 30, 30);
    exit();
  }
}
```

Appendix H **Embedded Device GreenAPP Data Transmitter Source Code**

Main program

```
#include <SPI.h>
#include <NewSoftSerial.h>
#include <Ethernet.h>
#include <string.h>

#undef int()
#include <stdio.h>
#include<stdlib.h>
#define SHARE_FEED_ID              35240    // Server Feed ID
#define UPDATE_INTERVAL            240000    //update frequency
#define PACHUBE_API_KEY
"qHO2LeYI1hKQZichkTP4rW7dbh5mPK9MVabYY9mD_0w" //API key

// Define the pins used for the software serial port.  Note that we won't
// actually be transmitting anything over the transmit pin.
#define rxP 8
#define txP 9
```

```
NewSoftSerial mySerial(rxP, txP);
char data[200];
double watts[9];
double tmpr;
int cnt=0;
int no=0;
int now[9];
String dataString="";
String dataT;
unsigned long last_connect=0;

int startPos=0;
int endPos=0;
char buff[64];
int pointer = 0;
byte mac[] = {
  0xCC, 0xAC, 0xBE, 0xEF, 0xFE, 0x81 }; // unique on LOCAL network
byte ip[] = {
  192, 168, 3, 177  };                  // no DHCP, set IP address
byte remoteServer[] = {
  173,203,98,29 };            // pachube.com
Client localClient(remoteServer, 80);

void setup()
{
  // set the data rate for the NewSoftSerial port
  mySerial.begin(57600);
  Serial.begin(57600);
  last_connect=millis()-UPDATE_INTERVAL;
  setupEthernet();
  initWatts();
}

void loop()                     // run over and over again
{
  readData();
  writeData();
}
```

Ethernet Connection Functions

```
void setupEthernet(){
  resetEthernetShield();
  Client remoteClient(255);

}
void resetEthernetShield(){
  Ethernet.begin(mac, ip);
}
```

Communication with remote server (Pachube)

```
int length;

void writeData(){
  if ((millis() - last_connect) > UPDATE_INTERVAL){
    if (localClient.connect()) {
      char  buf [20]= "";
     dataT = dtostrf(tmpr, 5, 1, buf);
      for(int i=0;i<9;i++){
        dataT += ",";
```

```
        dataT += dtostrf(watts[i], 5, 1, buf);
      }

      length = dataT.length();
      localClient.print("PUT /api/");
      localClient.print(SHARE_FEED_ID);
      localClient.print(".csv HTTP/1.1\nHost: pachube.com\nX-PachubeApiKey:
");
      localClient.println(PACHUBE_API_KEY);

      localClient.print("User-Agent: Arduino (Pachube In Out1.1)\n");
      localClient.print("Content-Type: text/csv\nContent-Length: ");
      localClient.println(length,DEC);
      localClient.print("Connection: close\n\n");
      localClient.println(dataT);
      initWatts();
      last_connect=millis();
      disconnect_Server();
    }
  }
}
void disconnect_Server(){
  localClient.stop();
  resetEthernetShield();
}
```

Reading the data from the energy monitor

```
void readData(){

  if (mySerial.available() > 163) {
    cnt = 0;
    while (mySerial.available()){
      char target = (char)mySerial.read(); //get character from serial
buffer
      data[cnt]=target;  //start filling the buffer
      cnt +=1; //move on to next character
      if ( target == '\n')  //if LF then we have a full line
        break;
    }//empty the NewSoftSerial copy to my buffer
    //verify data
    boolean nonsenseXML = false;

    for (int i = 0; i < cnt; i++) {
      int b = (int) data[i];
      if (b > 255) {
        nonsenseXML = true;
      }
    }

    if (nonsenseXML) {

      mySerial.flush();
      mySerial.end();
      mySerial.begin(57600);

      last_connect = millis();
    }

    //    Serial.print("data recieved: ");
    //Serial.println(data);
    int channel=convertStringtoInteger(parseSingleElement(data, "sensor"));

    if(no==0){
      tmpr=convertStringtoFloat(parseSingleElement(data, "tmpr"));
      no=no+1;
    }
    else {
      tmpr=((no*tmpr)+(convertStringtoFloat(parseSingleElement(data,
"tmpr"))))/(no+1);
      no=no+1;
    }
    if(now[channel]==0){
      watts[channel]=convertStringtoFloat(parseSingleElement(data,
"watts"));
      now[channel]=now[channel]+1;
    }
    else{

watts[channel]=((now[channel]*watts[channel])+(convertStringtoFloat(parseSi
ngleElement(data, "watts"))))/(now[channel]+1);
      now[channel]=now[channel]+1;
    }

  }//end of data available
}
String parseSingleElement(String input, String target){
```

```
  int start = input.indexOf("<"+target+">") + target.length()+2;
  int ends = input.indexOf("</"+target+">");
  return( input.substring(start, ends));
}

int convertStringtoInteger(String input){
  char ss[20];
  input.toCharArray(ss, sizeof(input));
  return atoi(ss );
}
double convertStringtoFloat(String input){
  char ss[20];
  input.toCharArray(ss, sizeof(input));
  return atof(ss );
}
void initWatts(){
  for (int i=0;i<9;i++){
    watts[i]=0;
    now[i]=0;
  }
  tmpr=0;
  no=0;
}
```

Appendix I **DVD Help**

Folder list

name	Function/purpose
GreenAPP-Computer-transmitter	Contains data transmitter source code and executable for Linux32, Linux64, macosx, windows64, windows32 and an Java applet. Source code can be run on Processing 1.5 included
Embedded-transmitter	Contains source code for embedded device transmitter Source code can be run on Arduino022
GreenAPP_demo	Contains GreenApp installed on a virtual mobile device. To open it, double click the shortcut with the same name. if short cut don't work run from command window emulator.exe with option @sami GreenApp can be found in applications once the virtual device starts If you find screen saver click the menu Once GreenApp start Click menu Choose change api key or feed ID Feed ID: 29649 API: qHO2LeYI1hKQZichkTP4rW7dbh5mPK9MVabYY9mD_0w
GreenApp Code	Has Green app code, runs on eclipse and need android SDK

processing-1.5.1	IDE
arduino-0022	IDE
Eclipse	IDE

Appendix J **Interviews transcripts**

Pre-trial Interview

Interviewer: (Introduces the topic)

Interviewee: Being able to see my electrical usage while I'm out of the house is okay, I think it's a very good idea, but if I'm worried about my meter usage there's very little I can do about it remotely to sort it out.

Interviewer: Currently there is little that can be done as part of the research being done is about intelligent homes and the next step naturally would be for your mobile app, if your appliance passes a certain threshold, to send a signal to the appliance to switch it off, but you need the right hardware to do this which is available from some companies but you need to prescribe a service, it's the same with people that provide home security.

Interviewee: Well I was thinking if I was away from home and I could be checking my electricity usage, I would feel powerless, there's nothing I could do about it so I'd rather not know.

Interviewer: That's a very good point but you know you could do something when you go home.

Interviewee: Do you know what I think would be a good thing, it sounds really silly but I think that there is a problem with water in this country as well as in your country. And in this country there's a problem with people setting up water irrigation systems in their gardens and, which are electrically powered. And if there is a mechanism able to stop it remotely it would be a good idea/ really helpful as they don't need to be on all the time, but one of the things that happens is people set them and leave them the whole day so

using much more water than they need. From an environmental point of view and the use of resources, this is waste.

Interviewer: Well this is achievable if you are very environmentalists and have a techy mind you can build this using micro-controllers. This is not available commercially but hobbyists do it all the time. The problem is not everybody would want to buy a micro controller and start programming it.

Interviewee: Yeah such a small section of society would do this. This is more for use of business application rather than home application - a person working in the council responsible for procurement having the ability to shut electricity down and calculating the cost of power consumption will be more useful for them.

Interviewer: There are a lot of companies that provide this for businesses and it is very expensive, proprietary not open source.

(Consent form signed at this point).

Are you doing anything currently to save energy to reduce energy consumption?

Interviewee: To reduce it, as long as I can remember we've been trying to use eco-cycles on the washing machine and dishwasher, certainly my partners much better on keeping an eye on things that are running. If I have something running will try and use it for multiple things rather than single things. I'm not sure it's saving energy but its making the most of what we currently spend. We've bought some insulation to save on heating which has worked very well. Again it's not something that we've done actively, but when we bought our house, we bought somewhere with double glazing to increase our energy efficiency. We haven't got a wind turbine or anything like that.

Interviewer: Well it's good, well I was just asking about things like turning lights off when you leave the room?

Interviewee: Sometimes.

Interviewer: Chargers?

Interviewee: Sometimes. There's a little thing on the computer that when you turn the computer off it turns off all the chargers.

Interviewer: Tumble drying?

Interviewee: We decided not to get a tumble-drier because they cost a lot of money, so not doing something doesn't feel like doing something.

Interviewer: Not buying a tumble-drier is a good conscious choice.

Interviewee: We got a microwave and those consume a lot of electricity.

Interviewer: That's not a bad thing they are useful. Do you leave appliances on standby?

Interviewee: Well generally I don't keep the TV on standby but I've got a virgin box that if I turn it off it needs a long time to boot so I leave it on standby.

Interviewer: I understand.

Interviewee: The other thing that I leave on standby is the computer, and we keep the mobile phones on the chargers so they're always recharging. Mobile phones, I think we need to do more about it because in our old house, all of our electrics were waist height so we could see if the switches were on or not, now in our new house the sockets are all near the floor so we don't notice as much. Some design elements collude with my laziness.

Interviewer: If you're making a cup of tea do you boil just the right amount of water?

Interviewee: Yeah, I use a coffee machine at home which only makes one cup at a time which probably consumes quite a lot of energy as we are two people at home.

Interviewer: Do you feel like you're doing enough to reduce your carbon footprint?

Interviewee: If 1 was the worst, and 10 was the best, I think we're doing about 4. We're not brilliant…This might sound silly again, but since the arrangements for our rubbish has changed with the council, we used to sort everything out, and now since they've changed, even the frequency of collection and ironically the council provided more

opportunities for recycling, we're doing less. When we used to have to go to the tip we used to do it more, it is that change at the point of when we stopped doing it as much.

Interviewer: What's the main motivator for you to save energy?

Interviewee: Guilt.

Interviewer: Guilt about what?

Interviewee: Guilt about using more resources than is fair, also making life difficult for other people, laziness stops me being efficient with energy not necessity. And we also use rechargeable batteries. This feels like really small stuff.

Interviewer: Small stuff adds up. Does saving money motivate you at all?

Interviewee: Yeah it does a bit, buying energy saving light bulbs doesn't feel like you're saving money.

Interviewer: Do you know exactly how much each appliance consumes at home?

Interviewee: Individually? I know how much a light bulb costs but I don't know how much energy it consumes.

Interviewer: Do you think participating in this trial will impact your energy consumption behaviour?

Interviewee: Yes, but I worry that it will be something like hawthorn effect, and I worry it's just the act of being watched that will make me use less electricity.

Interviewer: Do you think you could you be as honest as you can during the trial?

Interviewee: Yes, I will try. Recently I changed my mobile contract and I was on unlimited data before but now my data allowance is limited so I decided to use a mobile app to monitor my data usage on my mobile phone, and I can tell you now how much data I use every day and for every activity I do I know how much data I'm going to be using. So I think participating in this would have the same effect. In terms of the interface, different people respond to different things, I responded to the knowing if I continue to use it in a certain way I will have used 80% or 90%, so the trajectory of

using the data was the most helpful in terms of feedback.So rather knowing on a minute by minute basis, I prefer to know the end result, so on a daily basis.

Interviewer: Do you have an idea of your energy usage?

Interviewee: I'm not sure about the energy usage but I know how much it costs because when I got my contract with the provider, I tried to get the fixed rate as long as we consume a certain amount.

Interviewer: Given the fact that the average cost of an energy monitoring system is between £200 and £1000, will you ever consider purchasing one for your household? Why and why not?

Interviewee: Probably not, just on the basis of I would have to have clearer time frame of when I'd get the money back so for example my aunt has just bought a solar panel and prior to doing that she got some meters installed to estimate the time by which she will get her money back. So if you were to tell me that I spend £1000 on the device and save £2000 this year, of course I'm going to buy it. If I'm going to make that money back in 12 years time, it will not be as urgent, I'd be a bit more sceptical.

Interviewer: Would cheaper technology being available encourage you to get one?

Interviewee: Yeah. All new technology is expensive in the beginning, so if you wait a few years you will get it much cheaper. Unless you have a strong motivator to make you want to get that technology, you normally just wait until it's cheaper. If the device is given to me by the provider and the cost is hidden within my energy bill, I might be more inclined to get it.

Interviewer: So we've talked about the types of feedback you prefer, would receiving alerts be a good thing when you exceed targets?

Interviewee: I think on the application yes it would be useful, whereas if it was a text message or an email I would consider it junk mail and I wouldn't pay attention to it.

Interviewer: Do you find setting goals could be helpful like if you set a goal for yourself to spend a specific amount of money on your electricity or on a specific appliance, do you think that would be helpful?

Interviewee: If I set myself a goal then I start feeling anxious and worried about meeting it, I can see a trend and I can go with that trend, sometimes not setting goals is more effective than setting a particular goal, that's my individual psychology.

Interviewer: So if you see your energy consumption last month and this month is less, then this is a more helpful way of feedback?

Interviewee: Yes. So I may not say to myself I will set a goal of 20% reduction of energy consumption, but if I see a reduction I will try to continue with it. It's all to do with choice and control.

Interviewer: And if you see an increase happening will you try and do something to reduce it?

Interviewee: Yes.

Interviewer: Do you like the concept of comparing your energy consumption with other households? Like using the social network thing?

Interviewee: I think, unless I know they've got the same kit as me, not really.

Interviewer: So if there is supposedly an anonymous application that looks at the consumption of people that have the same appliances as you or shows you and compares you to them or shows you that you are doing better than those people, would that be useful?

Interviewee: Probably not.

Interviewer: Would an application that shows an average of consumption by other people be useful?

Interviewee: Averages are useful, I trust them more than individual statistics.

Interviewer: (Shows the images of figure 1)Do you like this kind of feedback or do you find other types of feedback useful?

Interviewee: That's good, this is one and zero, it's not for me, I prefer something that is graphical data. I think what would make more of a difference to me is seeing it in terms of money. Not because it is the thing of primary importance to me, but it still gives me a better idea of how I'm doing and if it's abstract I wouldn't know.

Interviewer: Would you like to see a carbon footprint i.e. CO2 Kg equivalent of your usage?

Interviewee: I don't think a CO2 footprint would be useful to me; I need a scale to understand what it means.

<u>Pre-trial Interview</u>

S.S: Are you currently doing anything to reduce your energy consumption?

G.L: Well, I Make sure I turn the lights off whenever I leave the room and I've been trying to use as little lighting as possible so we have some different lamps in the room and I'll try and just choose one to put on. I make sure I turn the television and laptop off when I'm not using them. I also have those speakers that stay on standby if you don't switch them off individually, you can actually hear them buzzing and using energy.

S.S: Things like do you leave appliances on standby or do you turn them off at the mains.

G.L: I never turn the TV off at the mains, and in my room, I just tend to turn it off at the laptop, does this mean that when you don't turn it off at the mains it is still using electricity?

S.S: No, standby is that you don't need time to boot it, so a laptop on sleep mode for example. Do you unplug chargers from the mains?

G.L: I leave my charger in the socket but the switch is off.

S.S: Do you have a tumble drier?

G.L: No

S.S: Do you feel you are doing enough to reduce your carbon footprint?

G.L: No probably not, I'm sure there's a lot more I could do, like have shorter showers, watch less TV, don't use computers and TV quite as much, I think there's a lot more I could do I just don't do it. And also flying to places I would find hard to give up as it's one of my main enjoyments in life.

S.S: What motivates you to save energy?

G.L: I'd say it is both; it used to be much more to do with the environment – It was something I felt very strongly about at the time when it was always in the media etc. about global warming and climate change. I was quite obsessed at one point with recycling so that there wasn't so much being consumed and wasted so less energy would be needed to produce new products. And then it kind of changed in my mind a little bit and the people around me weren't very motivated to do this, and I think the focus changed to money as well.

S.S: Do you think that participating in this trial will have an impact on your energy consumption behaviour?

G.L: Yes, will it be measuring gas as well?

S.S: No just electricity.

G.L: Yeah I think it might, as occasionally if I leave the TV on when I've gone upstairs and little things like that it would be interesting to see how much energy that uses.

S.S: Great. Do you have a clear understanding of your energy usage, do you like analyse your bills and things like that?

G.L: To see how much each appliance uses?

S.S: Yes

G.L: No

S.S: Is that because it's not in the front of your mind?

G.L: Yeah and because it's never been something that's occurred to me to do

S.S: When you buy appliances do you try and buy green appliances or do you understand what green appliances are?

G.L: Like low energy light bulbs and stuff like that? They're usually so much more expensive but I'll buy the low energy light bulbs where I can.

I also try and get wind up torched and solar powered gear that I can use for certain activities I like.

S.S: If you're buying a fridge for example, there are some that are greener than others.

G.L: Well the problem is because I'm renting houses every year, I don't really get that choice of using the appliances I would like. But I think if I get to the stage I can buy a house, I will make sure I try I buy the greenest things I could.

S.S: Do you carefully follow the environmental issues arguments and do you understand what a tonne of CO2 means?

G.L: I can't actually visualise what a tonne of CO2 would look like, I've no idea how much that would be, so no.

S.S: Given the fact that the average cost of a home energy monitoring system is between £200 and £1000, would you ever consider purchasing one for yourself? Why or why not?

G.L: I think I would, but I think I will know whether or not I would after this trial as

But I think if I had my own house and it only cost £170 then I would buy one and try and be responsible.

S.S: Do you understand the energy consumed by the different appliances in your home?

G.L: I have an idea that maybe certain appliances use more, so I would have assumed that a laptop would consume more than a lamp for example.

S.S: What type of feedback do you think will affect the amount of energy you use? It's like now you are consuming energy in your home, but you can't see how much you are consuming, so if there is a way of you seeing it, in what way would you prefer that?

G.L: Are there choices?

S.S: Yes like would you like to see numbers of what you are seeing now, or numbers comparing how much you consumed now and what you used last week, or would you like to see how much you are consuming in comparison with other people who have houses almost the same as yours?

G.L: I would like to see it for each appliance to see what you're using maybe on a daily basis and on a weekly basis. I think it would be good to have a function to see your energy usage at the end of the week and something that helps you set goals – saying you used this much electricity but that you could watch TV for one less hour and that can be your goal. But I'm not as bothered about comparing my usage to other people's usage.

S.S: (Showed the images of the app interface) do you think these images are helpful in helping you to save energy?

G.L: Yeah I think so, if I hadn't done very well that week, the image would show me if I'd done well or not.

S.S: Would it scare you into saving energy?

G.L: Yes probably.

S.S: Lastly, do you have any questions?

G.L: No that's great.

S.S: OK thank you very much.

Mobile Application Usability

How easily have you been able to navigate the application?

Pretty simple and intuitive. At first thought there were on screen buttons but soon worked it out. The process of entering the ID data and number was a bit complicated. Not sure i could do it again without a tutorial ;o)

How well did you understand the information in the application?

Ok. It's been a while since I used KWh – not something i have paid much attention to since high school physics. I understand the concept but it did not feature heavily in my

day to day thinking. Except now it does – a lot more. The graphs helped witht the information a lot.

Overall this application was easy to use

I would say for me quite easy, once it's set up (4/5) for ease

When using the mobile app "EMAPP", Could you obtain information to answer any queries you had regarding your energy consumption?

Yes, i'm not sure i had any queries i couldn't answer

Is the information displayed by the app timely and useful to you?

Very much so. It has raised my awareness significantly, and the ability to find information on my phone was very useful. I was over the moon to see billable use last month was only £22, and we have reduced our Direct Debit, and claimed s substantial amount of money back off our provider

Are there tasks that you will want to perform that are not currently supported by application?

No

Is the application performance fast enough for you?

Yes

Are the graphics used recognizable to you and do they facilitate ease of use/understanding?

Yes – see answer 2.

Can you successfully navigate through the application? Is the system status clear to you at all times? Is the navigation intuitive?

Yes, Yes, Yes. But it took a little practise (all apps take a little practice)

How do you rate the general handling of the app?

Not sure what this question means. Overall satisfaction was 4.5/5

What do you think of the colour scheme?

Great. I didn't notice it. Which means it did the job well. I focused on the data, not on the aesthetics

Energy Consumption Feedback

When using EMAPP, you energy consumption is feedback in power, monetary and CO2. Which is you preferred method?

I really depends. For a wider audience I think £. It's certainly the most usual measure for most people who aren't engineers/climatologists

Do you have any suggestion about the history feedback?

Not really.

Do you have any other suggestion regarding the type of information provided by EMAPP

I think I would have done better if I'd had chance to customise the name of the wall socket plus – to enable me to know that computer was "plug 3", and kettle was "plug 2" rather than having to remember plug 2 was "3" and plug 4 was "4". I sometimes got a bit confused

Appliance Energy Consumption

How did you feel about being able to monitor you appliance consumption?

It put me in greater control of my environment. It made me think more about consumption – although I'm not sure our consumption will have dropped (a third person came to live in the house ½ way through the experiment)

Do you think this experience will affect your future energy consumption behaviour?

Yes, I think it will. I would like to have lived with the app for a little longer though. But this is my own fault – no one else's.

Any other comments/ suggestions

None. Other than I use an app to monitor my data usage 3g watchdog. This dynamically calculates my usage, and predicts what I will look like at the end of the month/week/day. This tells me % of my allowance and predicts what it will look like in a month at my current levels. When I look it gives me a reading of my %of allowance used on either daily/weekly/monthly terms. This is a useful dashboard number to have as it allows me to alter behaviour immediately or hep me to attain goals. I recommend looking at 3g watchdog

Overall experience

Did you enjoy the experience?

Yes, very much. It was interesting to see things happening in the day.

Would you invest in an energy monitor?

It depends on the cost. Sorry not to be clearer. If i thought it would help me save the same amount ovr 3-5 years then yes

Any comments/suggestions

I was pleased to have been asked to take part. It was very good. I think the internet interface was a bit rough and I worried about how robust it was. Battery power would feel safer (if got rather hot off the mains at one point) and the exposed circuitry made me nervous of breaking it.